

PC WORLD
RUSSIA

МИР ПК

Журнал для пользователей персональных компьютеров

Анимация и компьютерные игры

Мобильные компьютеры

Windows: редактор PIF-файлов

Borland

Pascal 7.0:

новые

возможности

Программирование

видеоадаптеров

EGA и VGA

1'93





КОМПЬЮТЕРЫ ДЛЯ ПРОФЕССИОНАЛОВ

117192 Москва,
Мичуринский пр-т, 1,
тел. (095) 939-5975, 939-2201



Анимация и компьютерные игры

Мобильные компьютеры

Windows:
редактор
PIF-файлов



Журнал издается
Совместным предприятием
по информатике и вычислительной технике
«Информэйшн Компьютер Энтерпрайз»

PC WORLD
RUSSIA

МИР ПК

ОСНОВАН В 1988 ГОДУ

№ 1/93

Содержание

В фокусе

6 *Вы, будущее и мобильные компьютеры*

Мобильные компьютеры — это быстро развивающаяся технология, которая обещает изменить характер нашей работы и даже образ жизни. Яснее обозначить черты приближающегося будущего Вам помогут авторитеты компьютерного мира, давшие свои прогнозы развития мобильных компьютеров и их воздействия на нашу жизнь.

Кэтлин Вигнер

Аппаратные средства

12 *Кочевники девяностых*

Если у вас есть мобильный компьютер, вы можете сами решать, когда и где вам следует на нем работать.

Джефффри Ротфедер

17 *Лучше меньше, да лучше*

Как выбирать мобильный компьютер.

Анита Амиррежвани

19 *Мобильные компьютеры при восьми километрах в секунду*

Даже космический корабль, оснащенный мощными бортовыми компьютерами, уже не может обойтись без нескольких портативных ПК, способных решать сложнейшие технические задачи.

Филип Чен

20 *Пионеры информационной эры*

Наступает время, когда компьютер становится по-настоящему мобильным и общение с ним превращается в праздник, который всегда с тобой.

Брайан Хастингс

22 *Справочник покупателя*

Анита Амиррежвани

28 *Анатомия карманного и блокнотного компьютера*

Стив Басс



Компьютерные игры

- 31 **Показывать, чтобы увидели**
М.В. Донской

✓ 33 **Анимация в компьютерных играх**

Использование принципов диснеевской анимации поможет обеспечить комфортные условия восприятия экран-ных образов.

Алексей Орлов

- 42 **Своя игра**
Андрей Родионов

- 56 **Игры приносят успех? Да, если повезет.**

Интервью с директором отдела игр фирмы «ПараГраф» Георгием Пачиковым.

- 60 **Из жизни логических игр:
IV Международная компьютерная олимпиада**
И.В. Григорьев

Портрет фирмы

- 64 **Intel в Москве**

Трудно представить себе персональные компьютеры без микропроцессоров Intel. Еще труднее представить, что легендарный Эндрю Гроув оставит без внимания формирующийся российский рынок ПК.

И.Б. Рогожкин

Windows

✓ 70 **Таинственный редактор PIF-файлов**

Мы продолжаем публикацию журнального варианта книги Б. Ливингстона «Windows 3.1 Secrets».

Брайан Ливингстон

Форум

- 79 **Итоги конкурса «KEYMACRO.LEX—92»**

Благодаря хитроумным макрокомандам текстовый редактор ЛЕКСИКОН приобретает новые, удивительно мощные возможности.

Г.В. Шмерлинг

- 82 **Черные вести о Черном Мстителе**

Из-за Черного Мстителя нам, может быть, скоро придется вывешивать на компьютерах табличку:
«10.00—14.00: антивирусная профилактика».

Брайан Бранденбург



- 87 **Любители и профессионалы**

Первая из серии статей, посвященных проблемам ремонта и сервисного обслуживания ПК.

И.Б. Рогожкин

Ваш выбор

- 89 **Time Line 5.0: теперь с графикой!**

О новой русской версии системы управления проектами Time Line 5.0.

Сергей Гладков

Программное обеспечение

- ✓ 97 **Общая библиотека для Си и Паскаля**

Рассматриваемая в статье технология поможет программистам расширить список компиляторов, для которых предназначены создаваемые ими библиотеки.

Р.А. Нечаев, А.Н. Малярский

- ✓ 105 **Borland Pascal 7.0: новые возможности**

Описываются пакеты Borland Pascal 7.0 и Turbo Pascal 7.0, о выпуске которых на рынок фирма Borland объявила в ноябре 1992 г.

В.В. Фаронов

Практикум

- ✓ 115 **Как поставить точку**

На примере рисования точки рассматривается работа с видеопамятью в основных графических режимах адаптеров EGA и VGA.

В.Н. Чертков

Письма

126, 127

Новости

59, 114

В номере использованы материалы из журналов:
InfoWorld, MacWorld, Publish, Portable Office,
PC-Welt, PC World

Новые планы, новые надежды...

Новый год мы всегда подсознательно связываем с новыми планами и надеждами. Представляя читателям первый номер «Мира ПК» за 1993 год (номер некоторым образом юбилейный — 30-й со дня основания журнала в 1988 г.), хотелось бы рассказать о том, что планируется в нем опубликовать в текущем году и что делается, чтобы один из первых наших компьютерных журналов стал самым интересным и доступным.

Сначала о доступности. Все, кто не успел подписаться на журнал, могут сделать это через редакцию. Учтя, что многие читатели предпочитают сейчас покупать журнал в розницу, всех заинтересованных представителей регионов мы приглашаем к сотрудничеству по его распространению.

В этом году каждый номер «Мира ПК» будет иметь кроме постоянных разделов (Аппаратные средства, Новые продукты, Windows, Портрет фирмы, Практикум, СУБД и др.) дополнительный тематический раздел, более широко освещающий одну из таких тем, как компьютерные игры и анимация, графические оболочки, ЛВС, настольные издательские системы, телекоммуникация, мультимедиа, САПР, рабочие станции, UNIX, новые архитектуры и ряд других. Хотя большинство тем так или иначе «привязано» к внешним событиям, например, большим компьютерным выставкам, начали мы с «компьютерных игр», полагая, что эта увлекательная тема, безусловно, привлечет ваше внимание.

Читатель может также заметить, что в журнале увеличилось число статей отечественных авторов. Мы считаем, что при прочих равных условиях их статьи интереснее и ближе к нашей жизни.

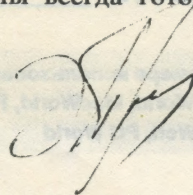
Каждый день с редакционной почтой нам приходят пресс-релизы, рекламные проспекты, фотографии и демо-дискеты продукции иностранных фирм. Получение же таких материалов от российских фирм — большая редкость. Между тем мы хотели бы иметь такую информацию: во-первых, она помогла бы нам лучше формировать обзоры по отечественным программным и аппаратным средствам, а во-вторых, позволила бы указывать контактные телефоны и факсы работающих на российском рынке, дистрибуторов и дилеров западных компаний в тех случаях, когда речь идет о возможности приобретения того или иного продукта.

В редакцию обратились несколько фирм с просьбой провести тестирование собираемых ими ПК, а также проверить качество русификации некоторых пакетов. Учтя опыт и поддержку тестового центра журнала PC World, мы организовали общественную тестовую лабораторию журнала «Мир ПК». Результаты тестирования будут, разумеется, публиковаться. Это позволит нам отмечать наиболее интересные продукты дипломом «Лучший выбор».

Еще один проект журнала связан с развитием некоммерческих форм распространения ПО, в первую очередь — с условно-бесплатным (Shareware). Совместно с недавно созданным Международным фондом «Общедоступные компьютерные программы» при журнале также организуется фонд общедоступного ПО. Фонд будет проводить сбор, тестирование, аннотирование и распространение программ отечественных разработчиков.

Для нас также очень важно, чтобы поиск новых форм не был односторонним, поэтому мы всегда готовы сотрудничать с Вами, читатель.

Главный редактор



Эдуард Пройдаков



ЖУРНАЛ
ПО ИНФОРМАТИКЕ
И ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ

Исполнительный директор
А.А. Константинов

Главный редактор
Э.М. Пройдаков

Научные редакторы
А.С. Рывлин
И.Б. Рогожкин
М.С. Суханова
А.И. Павловская

Художественный редактор
О.Д. Кузнецова

Литературные редакторы
Е.Н. Кудряшова
Н.К. Логинова

Директор службы рекламы
А.В. Лаврентьев

Корректоры
С.Ю. Бардина
Н.И. Лауфер

Оператор
Н.Х. Признякова

Операторы верстки
Н.Н. Лунькова
О.В. Царева

Художник обложки
С.Ф. Лухин

Подписано в печать 01.02.93. с готовых
диапозитивов фирмы «Мир ПК».
Формат 84×108¹/₁₆. Печать офсетная.
Тираж 1-го завода 30 000 экз. Заказ 37027.
Отпечатано в типографии акционерного
общества «Молодая гвардия».
Адрес АО: 103030, Москва, Суцеская, 21

Акционерное общество
«Информэйшн Компьютер Энтерпрайз»
адрес: 129223, Москва, пр-т Мира,
ВВЦ, ПОК
АО ICE
Верстка и оригинал-макет
изготовлен в АО ICE

2404000000—39
М
949(01)—93

Редакция: тел. 216-78-38
Отдел рекламы, распространения
и подписки: тел. 216-53-90

© Акционерное общество
«Информэйшн Компьютер Энтерпрайз», 1992
Полное или частичное воспроизведение или раз-
множение каким бы то ни было способом ма-
териалов, опубликованных в настоящем изда-
нии, допускается только с письменного разре-
шения Акционерного общества «Информэйшн
Компьютер Энтерпрайз»

Wonder lite

*Профессиональная
компьютерная техника
Всемирно известной марки "Wonder Lite"*

*компьютеры и сети,
периферия и комплектующие,
конфигурация по заказу,
программные продукты,
сервис и модернизация -*

За рубли и прямо в Москве!

Конечным пользователям, нуждающимся в опытным и надежном партнере - поставщике современной высококачественной техники, фирма **ProSoft** предлагает базовые модели PC AT марки **"WonderLite"** из Германии:

- 386/387 SX/DX 16-40 MHz, RAM 2-4 MB, HDD 80-120 MB, VRAM TVGA 512 kB - 1 MB,
- 486 33-66 MHz, 4-32 MB, Cash 256 kB, HDD 120-520 MB, TVGA 1 MB

Все модели оснащены: AT BUS, 2S1P, FDDs 5,25" 1.2 MB+3,5" 1,44 MB, SVGA 1024x768.

Профессиональные периферия и комплектующие:

- Принтеры - лазерные HP LaserJet II-IV, матричные Star, OkiML
- HDD: 80,120,210,330,520 MB
- Сканеры
- Звуковые адаптеры "SoundBlaster"
- Графические адаптеры VGA 256kB-1MB
- FDD 3,5" 2,88MB
- CD-ROM 630MB
- Контроллеры ESDI, SCSI, IDE-Cash, HDD, FDD
- Супермониторы EIZO, NEC
- Стримеры
- Материнские платы, процессоры, сопроцессоры
- Источники бесперебойного электропитания
- Потрясающие настольные боксы для дискет
- Профессиональные суперграфические видеоадаптеры HiColors (Windows, AutoCad ...)
- Картриджи для принтеров, стримеров
- Дискеты

Модернизация и усиление характеристик IBM PC

Консультирование и обучение персонала, установка локальных сетей.

Заключение договоров о сервисном обслуживании поставляемой техники с обеспечением расходными материалами, картриджами и пр.

Гарантия качества.

Особое внимание обратите на новинки (нам самим очень нравится):

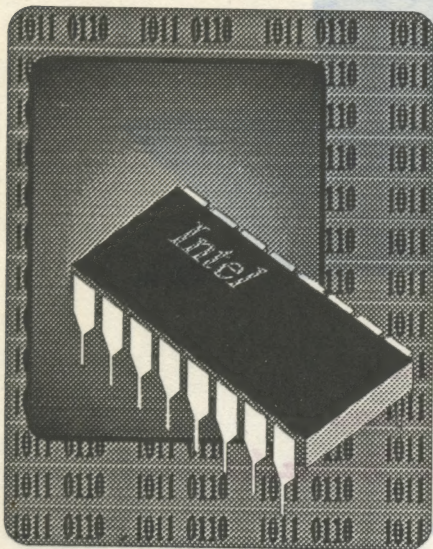
Съемный винчестер Установка в стандартный карман системного блока устройства, позволяющего простым поворотом ключа отключить и извлечь стандартный жёсткий диск с передней панели. Приобретение нескольких устройств позволяет оперативно заменять однотипные винчестеры в Вашем PC. Очень удобно и красиво.

SoundBlaster Pro Звуковой стереоадаптер для Вашего IBM PC AT. Входы: микрофон, линия, CD-audio; выход - на линейный вход Вашего стерео или наушники. Поддержка Windows, MIDI-стандарт. Программное обеспечение: CD-player, чтение текстовых файлов, диалог "вслух", проигрыватель музыкальных MIDI-файлов, звуковой редактор, аккомпанирующий орган, микшер и др. Игровые программы с полным стереозвук - просто потрясающе!

COMPUTER • PERIPHERALS • COMPONENTS

А/О "Прософт": ☐ 103012 Москва, Центр, Рыбный пер., д. 2, пом. 49 ☎ (095) 921-4333 факс, 298-3597, 921-1676

ProSoft



Вы, будущее и мобильные компьютеры

Кэтлин Вигнер

Мобильные компьютеры — это быстро развивающаяся технология, которая обещает изменить характер нашей работы и даже образ жизни. Яснее обозначить черты приближающегося будущего Вам помогут авторитеты компьютерного мира, давшие свои прогнозы развития мобильных компьютеров и их воздействия на нашу жизнь.

Говорить о мобильных компьютерах — значит прежде всего говорить о системах связи. Хотим ли мы постоянно поддерживать связь с фирмой или просто заниматься сбором данных, нам потребуются большие объемы разнообразной по форме информации: видео, речь и текст. Мобильные компьютеры будут похожи на универсальные бытовые устройства. С их повсеместным распространением могут уйти в прошлое транспортные пробки в часы пик, поскольку многие люди смогут полноценно работать у себя дома, у клиента или просто в дороге.

Некоторые авторитетные специалисты опасаются, что возможность заниматься работой, поддерживая связь в любом месте и в любое время, сотрет границу между работой и частной жизнью. Это вполне может произойти, но способность работать и управлять информацией в любом месте и потенциал, который заложен в этой свободе, обещает необычайно обогатить жизнь будущих поколений. Мы увидим электронные книги и виртуальные классные комнаты. Роди-

тели получают неоценимую возможность всегда знать, где их дети и чем они заняты.

Технология мобильных компьютеров развивается очень быстро. Важно не отстать и не остаться на обочине.

Филипп Кан (Philippe Kahn),
президент фирмы Borland
International



«Люди приспособливают инструменты к своим нуждам, — говорит Кан. — Сегодня многие хотят иметь возможность обратиться к работе в любое удобное для них время. По мере того как компьютеры становятся легче, меньше и дешевле, все больше людей могут поддерживать постоянную связь между собой. Скоро, куда бы вы ни

направились, вы сможете брать работу с собой». Кан утверждает, что в будущем улучшится и возможность контролировать работу. Чтобы оперативно посылать и получать сообщения по всему миру, совсем не обязательно созваниваться по телефону. Работать можно будет где и когда угодно. Повысится не только производительность труда, но и качество жизни. Транспортные проблемы также могут уйти в прошлое.

Джерри Каплан (Jerry Kaplan),
президент корпорации Go

«Основная цель мобильной компьютерной техники — поддерживать постоянную связь, — говорит Каплан, чья компания занимается разработкой программного обеспечения для машин с рукописным вводом. — Мне кажется, что в будущем обязательно появятся карманные индивидуальные связные устройства с наушниками и микрофоном. С их помощью можно будет передавать не только документы, но и речевые сообщения. Подобные устройства вытеснят факсимильную связь».

С развитием мобильной компьютерной техники продажа программного обеспечения переместится в сферу услуг. В компьютере будет предусмотрен режим дистанционного заказа программ, используемый по мере надобности. «Появится возможность создать даже систему дистанционного резервирования и использовать ее как удаленный архив», — размышляет Каплан.

Роберт Кэвнер (Robert Kavner),
президент корпорации AT&T

Термин «мобильная компьютерная техника» Кэвнеру не нравится. Он предпочитает говорить «коммуникационные устройства для мультимедиа».

«Идея миниатюрного персонального компьютера — это не самая удачная парадигма, — говорит Кэвнер. — Компьютерная техника не будет править новым миром».

Такие устройства позволят людям через единую связную систему поддерживать связь в рукописной, речевой и в видеоформах, причем послать подобное сообщение будет проще, чем письмо по почте. «Мы возродим на новом уровне старую традицию посылать открытки. Я пишу короткую заметку на экране-планшете, из встроенной записной книжки выбираю адрес и через сотовую сеть посылаю это сообщение моей матери. Почтовый ящик известит ее, что пришла открытка. Таким я вижу будущее».

Крэг Маккау (Craig McCaw),
исполнительный директор
корпорации McCaw Cellular

Маккау полагает, что «мобильные персональные помощники» (mobile personal assistants) позволят каждому общаться с миром наиболее удобным образом. «Каждый человек может завести несколько телефонных номеров или кодов доступа для различных целей, например для деловых партнеров, друзей и близких родственников, — полагает Маккау. — Так нужно защищаться от внешнего мира, который иногда вторгается в личную жизнь». Связь должна устанавливаться только с согласия обеих сторон, чтобы можно было или ответить, или просто принять сообщение.

Резидентные базы данных позволят людям во время путешествий получать необходимую информацию, например, расписание полетов, путеводители по городам и общие географические или исторические сведения. «Что чаще всего досаждают? — спрашивает Маккау. — Недостаток специфиче-

ской информации. Наша задача обеспечить быстрый и простой доступ к непрерывно обновляемой информации».

Ричард Шеффер (Richard Shaffer),
владелец компании Technologic Partners

Шеффер задумал создать прототип мобильного компьютера. Иногда он называет его технологическим пикомпьютером (Technologic Picocomputer), иногда — компьютерным «швейцарским армейским ножом». «В нем будет приемник системы персонального вызова, который позволит моей жене в любой момент связаться со мной и попросить заехать по дороге домой в магазин, — объясняет Шеффер. — В нем будет встроенный сотовый телефон и беспроводный факс-модем. В нем будет цифровая карта, фрагмент которой можно увидеть, задав название нужного места. Обратившись затем к глобальной навигационной системе, устройство укажет мне текущее местоположение и, возможно, подскажет, как добраться до места назначения. Еще было бы очень удобно всегда иметь под рукой и расписание полетов».

Шеффер полагает, что такое устройство может иметь множество конфигураций. Возможны разные сочетания, например сотовый телефон с телефонным справочником и дисплейным абонентским приемником или беспроводный факс и система электронной почты, совмещенная с записной книжкой.

Роберт Глейзер (Robert Glaser),
вице-президент корпорации
Microsoft по мультимедиа
и бытовым системам

Глейзер полагает, что мобильные компьютеры будут

развиваться в двух направлениях. Современные блокнотные компьютеры будут совершенствоваться и становиться более универсальными. Кроме того, появятся другие устройства, выполненные по той же технологии, однако покупатели не будут воспринимать их как персональные компьютеры. «Ясно, что подобные изделия будут отличаться от ПК, — говорит Глейзер. — Они будут очень маленькими, очень легкими, удобными для того, чтобы постоянно носить с собой, и будут иметь встроенные средства связи».

Глейзеру кажется, что важно не выплеснуть ребенка вместе с водой: «В мире создана обширная компьютерная инфраструктура, включающая базы данных, электронную почту, справочники, каталоги и календари. Нам нужно разработать удобный и простой интерфейс, который опирается на уже имеющуюся инфраструктуру».

Роберт Франкенберг
(Robert Frankenberg),
вице-президент корпорации
Hewlett-Packard

Мобильные компьютеры должны не «заваливать» людей информацией, а помогать им в любом месте и в любое время отбирать необходимые данные, полагает Франкенберг. «Если я хочу купить пальто, я могу вызвать каталог фирм L.L. Bean или Land's End. Чтобы найти ближайшую больницу или книжный магазин, будет достаточно нажать несколько клавиш. Это коренным образом изменит нашу жизнь».

Франкенберг предсказывает появление портативных приборов для видеосвязи, которые он представляет в виде планшетов: «Я могу говорить с вами, писать на своем планшете, и написанное будет появляться на вашем. Мы сможем обмени-

ваться картами, таблицами, чем угодно, причем это будет гораздо дешевле, чем видеоконференции. Подобные устройства будут не только обеспечивать взаимодействие между собой, но и одновременно черпать данные из других источников информации, так что не придется по несколько раз связываться заново».

Эндрю Гроув
(Andrew Grove),
исполнительный директор
корпорации Intel



Гроув считает, что сами по себе мобильные компьютеры не представляют никакого интереса: «Все их значение проявится при применении в бизнесе для расширения рабочего пространства». Вместо того, чтобы повсюду носить с собой один универсальный компьютер, люди станут пользоваться множеством различных небольших и легких специализированных карманных устройств. Эти устройства будут скорее напоминать периферийное оборудование, чем компьютеры общего назначения. Может возникнуть большое разнообразие таких устройств.

Примеры таких специализированных компьютеров уже есть — это карманные компьютеры корпорации Hewlett-Packard, предназначенные для работы с электронными таблицами, и карманные компьютеры, используемые в некоторых современных ресторанах для

передачи заказов на кухню. В процессе эволюции карманные устройства для деловых людей превратятся в очень легкие интеллектуальные устройства, предназначенные именно для связи. «Чаще всего я использую портативный ПК для передачи сообщений по электронной почте и через систему CompuServe, — говорит Гроув. — Поскольку для этого мне каждый раз приходится носить с собой около трех килограммов лишнего веса, я считаю, что будущее именно за устройствами узкого назначения».

Николас Неграпонт
(Nicholas Negraponte),
директор Лаборатории средств
информации Массачусетского
технологического института

Неграпонт рекомендует перекладывать решение некоторых рутинных задач «на плечи» интеллектуальных устройств, которые он называет «агентами». «Дайте карманному устройству всю необходимую информацию, и оно сможет принимать решения без предварительных консультаций: в каком ресторане заказать ужин, с какими абонентами соединять, на какой авиарейс зарезервировать билеты. Это как при поручительстве, — говорит Неграпонт, — чтобы ваш «агент» мог принимать хорошие решения, будьте готовы раскрыть ему некоторые личные пристрастия».

Джон Эванс (John Evans),
исполнительный директор
корпорации News Electronic Data

Экспериментальная компания News Electronic Data была создана фирмой News Corporation Limited для обеспечения пользователей данными в дороге. Эванс считает, что в будущем большой популярностью будут пользоваться местные новости. «Нам кажется, что

новости тем ценнее, чем они локальные, — говорит Эванс. — О событиях в мире можно узнать по каналу CNN или USA Today. Ненадолго приехав в другой город, человек более озабочен тем, что происходит в его родном городе, а не там, куда он приехал». Занимаясь изданием газеты Village Voice, Эванс большое внимание уделял развитию раздела частных объявлений. Он считает, что спрос на частные электронные объявления будет весьма высок. «Сейчас это трудно представить: виртуальное представление действительности, когда «электронные агенты», а не люди, будут встречаться друг с другом». Если «агенты» найдут общий язык, встреча хозяев скорее всего пройдет успешно.

**Даррелл Миллер
(Darrell Miller),**

исполнительный вице-президент
корпорации Novell

Миллер термину «мобильная компьютерная техника» предпочитает название «кочующая компьютерная техника» и считает, что придет время, когда мы будем на записях носить связные телевизионные устройства в стиле Дика Трейси. Тогда в ближайшем будущем самым главным ограничением для «кочующей компьютерной техники» станет срок жизни батарей. «Из-за этого, — говорит Миллер, — переносные устройства не могут иметь вычислительную мощность полноразмерного компьютера. Здесь свое слово должна сказать сетевая технология. Обработка информации будет осуществляться в сети, а пользователю будет высылаться лишь результат». Фирма Novell уже разработала прототип местной сети, которая позволяет поддерживать постоянную беспроводную связь на всей территории завода этой фирмы в г. Прово (шт. Юта).

Джон Раймер (John Reimer),
основатель и президент
ассоциации PCMCIA

Раймер видит будущее как мир гнезд. Не осинных гнезд, а гнезд стандарта PCMCIA для плат расширения (см. «Мир ПК», № 10/92), которые обычно называют PCMCIA-картами. В виде таких карт реализуются различные устройства для хранения и передачи данных. Гнезда стандарта PCMCIA встраиваются в самые новейшие модели субблокнотных компьютеров. Раймер считает, что гнезда для PCMCIA-карт скоро появятся в телефонах в аэропортах и отелях и в торговых автоматах. Через эти гнезда можно будет получить доступ как к базам данных, содержащим географические карты, сведения об экскурсиях, адреса отелей, новости, так и, конечно же, к вашему банковскому счету. Выбрав необходимую информацию, вы загрузите ее в карту памяти. Счет за услугу будет выставлен автоматически. С помощью PCMCIA-карт можно будет подключаться к телефонным линиям. «Информацию можно загружать ночью во время зарядки карманного устройства, — говорит Раймер. — К утру все будет готово».

**Джек Блуменштейн
(Jack Blumenstein),**

президент корпорации Ardis

Корпорация Ardis была создана для работы в области беспроводных ЛВС. В настоящее время беспроводными сетями чаще всего пользуются выездные сервисные службы и некоторые руководители с соответствующей спецификой работы. Однако в начале 1994 г., по утверждению Блуменштейна, ожидается расширение области применения и круга пользователей подобных систем.

Кроме того, среди новых задач беспроводных ЛВС будет

оказание услуг по передаче сообщений потребителю, передача информации о курсах акций и дистанционное управление домашними устройствами. «Мы будем развивать специализированные сети, — говорит Блуменштейн, — а портативные связные устройства смогут пользоваться любой из них».

Пол Джен (Paul Jain),
исполнительный директор
корпорации Media Vision

«Исправить недостатки мобильных компьютеров помогут средства мультимедиа, — говорит Джен. — Они позволят применять мобильные компьютеры для домашних нужд и целей образования». Дети, отправляясь на каникулы, смогут взять с собой игры, домашние задания, книги, музыку и рассказы экскурсоводов. Все это поместится в карманном мобильном компьютере. Джен очень хочет, чтобы мобильный компьютер помогал ему отслеживать расписание спортивных мероприятий его сыновей: «Я никогда не могу определить, где они. Было бы полезно, если бы в школе мне дали расписание, которое я бы носил в мобильном компьютере, а он голосом говорил бы мне о том, что происходит».

Рэй Брэдбери (Ray Bradbury),
писатель-фантаст



Брэдбери, который пишет не на компьютере, а на старой пишущей машинке производ-

ства корпорации IBM, беспокоится, что мобильная компьютерная техника так перегрузит людей, что у них останется очень мало времени для творчества.

«Я недавно летел на самолете, — говорит Брэдбери, — два бизнесмена, сидевшие справа и слева от меня, были увешаны разными устройствами. Один из них попросил меня поменяться с ним местами, чтобы он мог сесть рядом с коллегой, но я отказался. «Вам нужно время для отдыха, — сказал я. — В самолете хорошо размышлять над философскими вопросами. Нельзя постоянно работать с людьми». Когда мы выходили, он поблагодарил меня».

Нобуо Мии (Nobuo Mii),
вице-президент и генеральный
директор фирмы Enter System
Technology (корпорация IBM)

Мии полагает, что мобильная компьютерная техника расширит область применения компьютеров, привлекая лю-

дей, которые ранее никогда ими не пользовались. Например, с помощью планшета с рукописным вводом страховые агенты смогут набросать картину повреждений машины и подсчитать убытки прямо на месте. На портативном принтере тут же будет напечатан чек.

«Мы также разрабатываем целый спектр новых мобильных коммуникационных устройств для выездных служб, — говорит Мии, — для автомобильных перевозок, служб доставки и сервиса». Каждый, кто хочет постоянно поддерживать связь, сможет иметь «информационного партнера». С помощью такого устройства можно вызвать компьютер вашей конторы из аэропорта в Сан-Франциско, сверить график встреч, изменить заказа на авиабилеты, прочитать электронную почту и даже созвать совещание. После всего этого можно по встроенному сотовому телефону позвонить домой и предупредить, что вы опоздаете к обеду.

Майкл Делл (Michael Dell),
исполнительный директор
корпорации Dell Computer

Корпорация Dell недавно представила первый промышленный полнофункциональный портативный компьютер на базе процессора 386SL, весящий менее 2 кг. Однако Майкл Делл считает, что мобильные компьютеры будут чем-то значительно большим, чем просто малогабаритные компьютеры: «Возникнет множество устройств, но никто не будет воспринимать их как компьютеры. Они будут предназначены в первую очередь для связи».

Эти устройства в любом месте и в любой момент помогут получить доступ к необходимой информации. «Многие люди зависят от информации, — говорит Делл. — Им нужен быстрый доступ к данным. Это потребует объединенных усилий компьютерных фирм, поставщиков информации и связанных компаний». ♦

INTERLINK

**386, 486 компьютеры з-да SIREX, USA !
И любая периферия за 15 дней из США.**

COMPUTERS

FILE SERVERS
486DX/33 EISA
486DX/33-50 ISA
386DX/25-40 ISA
386SX/25-33 ISA
SPEEDMASTER 486DX
WORKMASTERS 386DX
TASKMASTER 386SX

HARD DISKS

HD DRIVE 52-240 MB
HD DRIVE 300-667 MB
HD DRIVE 1.2 GB
HD DRIVE OVER 1.5 gb

COPROCESSORS

CO-PROCESSOR 387/25
CO-PROCESSOR 387/33
CO-PROCESSOR 387/40
CO-PROCESSOR WAITEK

PRINTERS

COLOR PRINTERS
DOT MATRIX PRINTERS
INK JET PRINTERS
LASER PRINTERS
THERMAL PRINTERS
FONT CARTRIDGES
TRACTOR FEEDERS
PRINT EXPANSION

ACCESORIES

CONVERTERS
UNTERRUPTIBLE PS
CABLES
CONNECTORS
MEMORY CARDS
JOYSTICKS

MONITORS

MULTISCAN MONITORS
SUPER VGA MONITORS
MDA MONITORS
HIGH RESOLUTION
PAPER / WHITE
VGA MONOCHROM

I/O DEVICE

EXTERNAL MODEM 9600
EXTERNAL MODEM 4800
EXTERNAL MODEM 2400
INTERNAL MODEM 9600
INTERNAL MODEM 4800
INTERNAL MODEM 2400
FAX MODEMS
I/O BOARDS
PLOTTERS
SCANNERS
NETWORK
DIGITAZERS
CD-ROM
FLOPPY DISK DRIVE
OPTICAL STORAGE

Информация: (095) 456-66-44, 453-10-96

Москва, Флотская ул., дом 13, корп.4. Факс: 456-83-69.

Intuitia

for
Windows -
программа
автоматического
распознавания
текста

Рабочее место для автоматического ввода печатных текстов с бумажных носителей.

Функциональное назначение: ввод и распознавание текста, спасение в виде файла в формате одного из текстовых редакторов с сохранением верстки, редактирование, проверка орфографии, верстка и вывод переработанного текста на принтер.

Аппаратные средства: IBM-совместимый персональный компьютер 386SX с 4 MB RAM, планшетный сканер HP ScanJet, лазерный принтер HP LaserJet.

Программное обеспечение: программа Intuitia, Windows 3.0 и старше, программа DeskScan, текстовый редактор (Word, Write).

Мультизадачный, параллельный режим работы.

Одновременно обрабатываются три страницы: одна сканируется, вторая распознается, а третья - редактируется.

Работа с бланками. Введенный со сканера бланк разбивается на области интереса, расположение которых сохраняется в библиотеке бланков.

Телефон: (095) 924-5811

Факс: (095) 924-3775

103051 Москва; а/я 7

E-mail: info@itco.msk.su



АйТи

Hewlett-Packard Authorized Dealer

Кочевники девяностых

Джеффри Ротfeder

Если у вас есть мобильный компьютер, вы можете сами решать, когда и где вам следует на нем работать.

Тот, кто смотрел летние Олимпийские игры 1992 г. в Барселоне, мог обратить внимание на 24-летнюю Мэри-Лу Реттон, первую гимнастку из США, завоевавшую олимпийское золото (1984 г., Лос-Анджелес). Она присутствовала в Испании в качестве обозревателя газеты *USA Today* и впервые пользовалась блокнотным ПК как для того, чтобы писать статьи, так и для того, чтобы отсылать их в газету с помощью новейших средств электроники. «Сначала я немного терялась, — говорит Реттон, — и каждый раз, когда сотрудники *USA Today* в Испании видели мое приближение, они разбегались и прятались — дескать, вот она снова со своими дурацкими вопросами».

Нельзя сказать, что Мэри-Лу компьютерофоб. У нее дома есть компьютер, и она использует его для подготовки текстов речей и для контроля за личными расходами. Но блокнотный ПК, по ее словам, — это нечто совсем другое: «Возможность повсюду брать с собой компьютер рождает чувство всемогущества».

Реттон быстро научилась пользоваться этим могуществом и вскоре с легкостью обращалась с блокнотным ПК. Каждый вечер в гостиничном номере она стучала по клавиатуре, печатая свою ежедневную колонку; затем она подсоединяла

машину к телефонной розетке, нажимала несколько клавиш и через модем мгновенно отправляла подготовленный материал в редакционные и наборные службы газеты *USA Today* в Испании и Вирджинии. «Я не знаю, как все это работает, — говорит Реттон, — материал отправлялся без всяких помех».

Реттон правильно определила основное преимущество мобильных компьютеров: еще несколько лет назад было невозможно даже представить, что люди самых разных профессий смогут повсюду носить с собой маленькие и легкие, но очень мощные компьютеры.

ВЕСТИ ДЕЛО ПО-НОВОМУ

Мобильные компьютеры (портативные, блокнотные, субблокнотные, карманные) получили за короткое время такое распространение, что изменили собственно характер труда людей. Возьмите, например, 65 распространителей издательства *Condé Nast Publications*, которые ездят по всей Америке с блокнотными ПК корпорации Toshiba для обсуждения с оптовиками и продавцами газетных киосков проблем продажи периодических изданий и предстоящей подписки. После того как агенты по продаже за-

несут полученные сведения в свои компьютеры, они пересылают информацию по телефонным линиям в штаб-квартиру компании в Нью-Йорке, где она немедленно анализируется и по результатам анализа принимаются соответствующие действия. Этот процесс протекает значительно быстрее, чем подготовка рукописных отчетов о переговорах, единственного прежде способа передать в центральный офис информацию о том, что происходит на местах.

Или, скажем, Шон Бернс, 28-летняя исследовательница политики в области здравоохранения из Атланты, которая год назад стала использовать блокнотный ПК корпорации Sharp с ЦП 286, чтобы упростить сбор информации в медицинских библиотеках, где она проводит большую часть своего дня. «Компьютер сэкономил мне уйму денег, так как отпала необходимость в фотокопировании», — говорит она. Бернс объезжает 159 округов в штате Джорджия для определения там уровня здравоохранения. Благодаря мобильному компьютеру она может записывать результаты прямо в ходе поездки и получать важную информацию простым нажатием нескольких клавиш. Более того, теперь ей по возвращении не придется расшифровывать сотни страниц неразборчивых заметок. «Нет ничего лучше, чем быть организованной, когда весь мир вокруг вас в беспорядке», — говорит Бернс.

Несмотря на это, Бернс обеспокоена тем, что привлекательность мобильных компьютеров может изменить ее жизнь, причем далеко не в лучшую сторону. «Мобильный компьютер столь удобен, что трудно удержаться от того, чтобы не брать его с собой снова и снова, превращая свою работу в нечто бесконечное. Но нужно

вовремя отложить в сторону эту «отраву».

Однако если некоторым компаниям удастся достичь задуманного, вам трудно будет отказаться от мобильных компьютеров, как от хорошей книги. В качестве эксперимента около дюжины издательств выпустили серию мультимедиа-книг, которые можно читать только на блокнотных ПК Powerbook фирмы Apple Computer. Среди них доисторические приключения «Юрский заповедник» (Jurassic Park) Майкла Кричтона, причем здесь не только изображается текст, но добавлены также звук и движение, когда оживают на экране описанные в романе динозавры.

ЭВОЛЮЦИЯ

Мобильные компьютеры прошли путь от забавных игрушек до привычных предметов очень быстро. И не удивляет предположение президента корпорации Lotus Development Джима Манзи, что «они будут определять передовые рубежи в течение следующих пяти лет».

В действительности это происходит уже сегодня. При том что размеры и стоимость электронной «начинки» быстро уменьшаются, каждое новое достижение в области конструирования мобильных компьютеров приводит к такой быстрой замене старых моделей, что вчерашний портативный компьютер — устройство весом 14 кг с питанием от батарей, которое приводило всех в восторг в середине 1980-х годов, — по сравнению с сегодняшней электроникой выглядит паровым котлом. И скорость таких изменений только возрастает. Совсем недавно портативный ПК размером 36 × 36 см и весом 4,5 кг или больше был последним до-

стижением техники. Но его уже превзошел блокнотный ПК, значительно более удобный в управлении, размером 20 × 28 см и весом 1,8—3,2 кг.

Цифры продаж подтверждают эту тенденцию. Между 1990 и 1993 гг. ежегодная продажа портативных ПК, по прогнозам исследующей рынок корпорации International Data (IDC) (Кеймбридж, шт. Массачусетс), возрастет на 53% и достигнет 288 тыс. штук. В течение того же периода, предсказывает IDC, ежегодные закупки блокнотных ПК взлетят на 254% и достигнут 1,2 млн. штук. Более того, блокнотные ПК, вероятно, недолго будут иметь такой высокий спрос. Мы уже видим появление уменьшенных вариантов, так называемых субблокнотных ПК. Разработанная изготовителем клонов IBM PC корпорацией Dell Computer модель 320SLi имеет размеры 18 × 28 см и вес всего лишь 1,6 кг.

Компьютерные компании не удовлетворены даже этой новинкой. Все известные своей передовой технологией фирмы (среди них IBM, Apple, Toshiba, Hewlett-Packard, NCR и Compaq), обгоняя друг друга, из месяца в месяц предлагают все новые и новые модели, каждая из которых имеет что-нибудь свое для привлечения покупателей. «Они не отрицают очевидное — то, что в ближайшие годы будут потрачены значительные суммы на мобильные компьютеры», — говорит Мэтью Кейн, руководитель программы маркетинговой фирмы Meta Group (Стэмфорд, шт. Коннектикут).

НОВЫЕ ВОЗМОЖНОСТИ

Борьба за захват этого растущего рынка становится

ся все напряженнее по мере усиления функциональных возможностей и уменьшения размеров машин. Находящиеся на гребне технологического прорыва, мобильные компьютеры уже сейчас ничем не уступают настольным. В течение примерно года самые последние модели портативных и блокнотных ПК были оснащены оптимизированными версиями микропроцессора (МП) 386 корпорации Intel, предназначенными для работы при более низких уровнях потребления энергии. Уже первые мобильные компьютеры, построенные на основе последнего поколения МП 486, сравнимы с большинством компьютеров, изготавливаемых фирмами AST, Toshiba и NEC. Более того, новейшие блокнотные ПК предлагают яркий, цветной экран размером 25 см — больший, чем экран настольного ПК Apple Macintosh — и более «легкий» для глаз, чем любой цветной монитор PC.

Но, пожалуй, самое замечательное — это размеры оперативной и внешней памяти, помещаемой в миниатюрные коробки. Обычно портативные и блокнотные ПК выпускаются с 2-Мбайт оперативной памятью и 60-Мбайт жесткими дисками; этого более чем достаточно для работы со стандартными прикладными пакетами, такими, как текстовые процессоры, электронные таблицы, СУБД и программы, работающие в графическом режиме. А если 60 Мбайт оказывается недостаточно (некоторые говорят, что это слишком мало для эффективной работы полного набора программ, одновременно использующих популярную операционную среду Windows), то сейчас уже появились на рынке блокнотные ПК с накопителями емкостью более 200 Мбайт. Такой диск может вместить до 100 тыс. страниц текста. Его

емкости не только достаточно для работы Windows, но и для требующих много дисковой памяти программ типа настольных издательских систем.

Кроме использования технологии последовательного подключения к источнику питания, большинство портативных компьютеров предполагает также оснащение существенными дополнительными устройствами, например модемами и адаптерами сотового телефона и ЛВС. Эти устройства позволяют посылать и принимать данные через телефонные линии и другие средства связи. Именно возможность повсюду носить с собой огромные объемы информации в маленькой легкой упаковке и обмениваться данными с другими машинами делает мобильные компьютеры столь привлекательными.

Фернандо Гельбард — бывший посол Аргентины во Франции — один из тех, кто довел до совершенства «мобильно-компьютерный» образ жизни. Хотя Гельбард возглавляет крупнейшее в Аргентине промышленное объединение Grupo Fate-Lauar, изготавливающее покрышки и алюминий, он едва ли когда-нибудь заглядывает в штаб-квартиру в Буэнос-Айресе. Он живет в Беверли-Хиллз и совершает длительные деловые поездки по всему миру, во время которых он управляет компанией через блокнотный ПК 386 Compaq LTE. Как только Гельбард прибывает в отель, он подсоединяет блокнотный ПК к телефонным линиям, читает корреспонденцию в электронном почтовом ящике и затем посылает ответы, напоминания и указания компаньонам или своей дочери в Калифорнии. Когда те, кому направлено сообщение, находятся далеко от компьютера, Гельбард посылает факсимильное сообщение. Его трехлетняя внучка, которая не умеет читать, очень доволь-

на, когда получает от него факсимильные изображения с понятными ребенку картинками. «Блокнотный ПК делает мою работу совсем неформальной», — говорит Гельбард.

Желающие приобрести мобильный компьютер со сложными возможностями связи и мощными процессорами должны знать, что модели наивысшего уровня оказываются недешевыми. Полностью укомплектованный блокнотный ПК с цветным экраном, жестким диском на 120 Мбайт и самым современным процессором 486 может стоить до 6000 долл. Для сравнения: настольная версия той же машины стоит 2000 долл.

Аналитики говорят, что если и существует одно-единственное препятствие для предсказанного стремительного роста рынка, то это необходимость быстрого понижения цен. «Две тысячи долларов — это тот максимум, который многие люди действительно готовы отдать за компьютер», — говорит Брюс Стефен, старший исследователь в IDC. В 1991 г., замечает Стефен, 43% всех проданных ПК стоили от 1000 до 1999 долл. В 1990 г. только 32% были в этом диапазоне цен.

КОНКУРЕНЦИЯ

Изготовители мобильных компьютеров реагируют на возрастающую жесткость конкурентной борьбы, перепроизводство машин и удешевление электронной «начинки». Хотя цены на эти компьютеры остаются относительно высокими, в середине 1992 г. они были уже на 20% ниже, чем в начале года. И не ожидайте, что эта тенденция изменится — в компьютерной промышленности это правило: по мере роста производи-

тельности цены падают. Если бы автомобильная промышленность развивалась такими же темпами, что и компьютерная, мы сегодня смогли бы покупать автомобили, которые расходовали бы на 100 км меньше литра топлива и стоили бы меньше доллара.

Даже при быстром снижении цен портативные и блокнотные ПК, вероятно, будут в течение длительного времени, по крайней мере по ценам, находиться в тени карманных компьютеров, новейшего пункта в компьютерном каталоге. Эти машины стоят всего 300—700 долл. и столь малы (размером примерно 15 × 3 см и весом менее 450 г), что помещаются в нагрудный карман. Несмотря на свои миниатюрные размеры, многие карманные компьютеры — такие, как «бестселлер» HP-95 LX компании Hewlett-Packard — являются полномасштабными мини-PC, с той же самой операционной системой DOS, что и в большинстве настольных компьютеров. Это значит, что на них можно работать с текстовыми процессорами и электронными таблицами, а данные могут быть переданы на более крупные ПК.

КАРМАННЫЕ КОМПЬЮТЕРЫ

И все же карманные компьютеры не могут полностью сравниться с ПК. С одной стороны, миниатюрными клавиатурами трудно пользоваться, с другой — их ОЗУ недостаточно для исполнения более крупных программ. Кроме того, у выпускаемых в настоящее время карманных компьютеров отсутствует подсветка экрана. Поэтому единственное назначение большинства из них — использование в качестве личных

информационных систем, комбинирующих записную книжку, дневник, цифровую адресную книгу и календарь со звуковым сигналом, напоминающим о назначенных встречах.

Карманные компьютеры продаются довольно хорошо: только в 1992 г. было куплено около 600 тыс., как утверждает Тим Байарин, президент консультационной фирмы Creative Strategies из Силикон-Вэлли. Но круг покупателей оказался не столь широк, как надеялись изготовители. Он охватывает прежде всего тех пользователей компьютеров, любителей техники и специалистов фирм, которые явно нуждаются в таких изделиях. Среди последних — лондонский ресторан Rules, где официанты и официантки с помощью электроники передают заказы на кухню с карманных терминалов, и фирма Frito-Lay, водители которой используют карманные компьютеры для отсылки в офис данных о продаже для своевременного пополнения запасов.

Такого рода особенности «демографии» покупателей должны ярко проявиться с освоением нового поколения карманных компьютеров — таких, как Newton корпорации Apple, Information Assistant корпорации IBM и карманные компьютеры фирм NCR, Hewlett-Packard, Sharp, NEC и Grid, — которые как раз сейчас поступают на склады или поступят в

начале 1993 г. Большинство аналитиков предсказывает, что эти машины будут популярны в 1990-е годы. «Ближайшие новинки в области карманных компьютеров могут изменить ежедневные привычки так же, как кассетный видеомagnetofон и сотовый телефон изменили их в 1980-х», — говорит Марти Мэнкин, редактор еженедельника *Take It With You* (Орем, шт. Юта), который следит за новинками в мире мобильных компьютеров.

ПЕРВЫЕ КОМПЬЮТЕРЫ

С первого взгляда ясно, что отличает эти новые карманные компьютеры и делает их более привлекательными для покупателей. Вместо стучащей клавиатуры многие предлагают похожее на авторучку устройство для ввода команд и даже текста. Это позволяет «объяснить» компьютеру, что делать, так же просто, как отметить пункт в списке. У некоторых есть даже немного «интеллекта»: имеется в виду, что, если вы достаточно часто ищете номер телефона Джима Смита, через некоторое время вам нужно будет только дать задание компьютеру «найди Джима» — и он выйдет прямо на Смита.

Все это очень удобно, но что действительно выделяет появляющееся поколение мобиль-

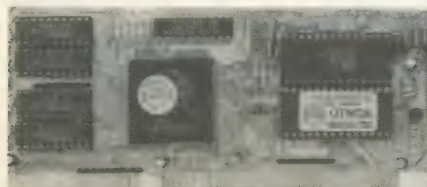
ных компьютеров — так это беспроводные средства связи.

Сотовая и радиосвязь позволит пользователям мобильных компьютеров получать и посылать разнообразную информацию, такую, как электронная почта, факсимильные изображения, отчеты и электронные таблицы. Можно будет также загружать информацию служб, ведающих новостями, прогнозом погоды, результатами спортивных соревнований, биржевыми курсами и официальными правительственными бюллетенями. И по мере того, как все больше компаний будет подсоединяться к этим сетям, люди смогут оплачивать счета, покупать товары и ценные бумаги и переводить деньги со счета на счет в любое время, в любом месте.

Билл Гейтс, президент корпорации Microsoft, крупнейшего изготовителя программного обеспечения для ПК, называет следующее поколение карманных компьютеров — «компьютер-бумажник», потому что они заменят наличные деньги, ключи, кредитные карточки, адресные книги, дневники и все остальное, что люди носят с собой каждый день. Когда это произойдет, мобильные компьютеры станут информационными прикладными системами, используемыми всеми. С их помощью уже сейчас стало возможным носить свой офис с собой — независимо от того, куда вы идете. ♦

ELSI LTD

ЛЮБЫЕ КОМПЛЕКТУЮЩИЕ ДЛЯ ВАШИХ КОМПЬЮТЕРОВ



Внешнеторговая фирма "Элси" предлагает со склада в Москве:

СИСТЕМНЫЕ ПЛАТЫ
ДИСКОВОДЫ
КЛАВИАТУРЫ
ПРИНТЕРЫ

ВИНЧЕСТЕРЫ
МОНИТОРЫ
КОРПУСА
МОДЕМЫ

ПЛАТЫ ARCNET И ETHERNET

Контрактные поставки из США
и стран Юго-Восточной Азии

Москва, Ленинский проспект, 35-А

Тел. 952-02-18
952-02-38

Факс 976-52-80
480-22-38

МИР ПК № 1/93 15



КВАЗАР • МИКРО

Официальный дистрибьютор корпорации Intel,
официальный реселлер фирмы Epson и корпорации APC

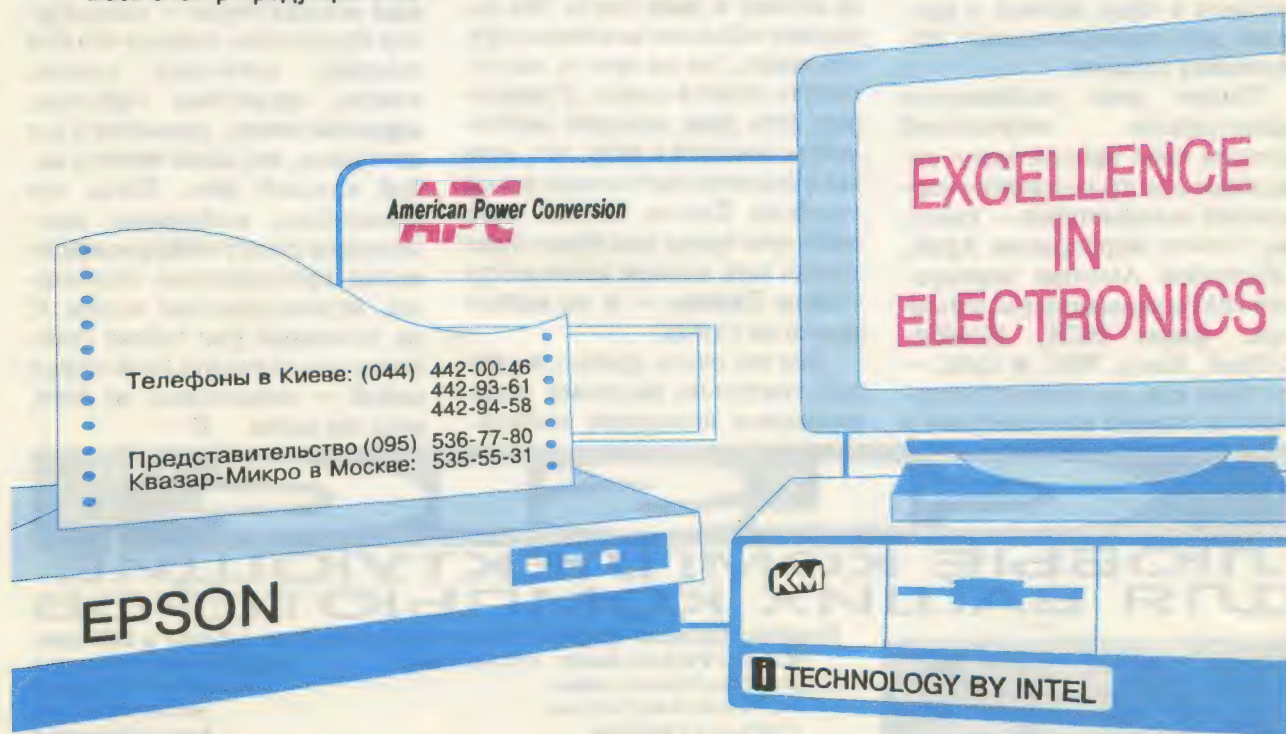
КВАЗАР-МИКРО -

- это совершенство в электронике

и надежность в партнерских отношениях

- Разработка цифровых и аналоговых микросхем в кратчайшие сроки, их серийное производство
- Отладочные средства для разработки систем на основе компонентов Intel
Изготовление модемов и факс-модемов на основе компонентов Intel
- Компьютерные системы на заказ, установка компьютерных систем
Весь спектр продукции Intel

- Источники бесперебойного питания корпорации APC
Матричные, струйные и лазерные принтеры фирмы Epson



Лучше меньше, да лучше

Анита Амиррежвани

Как выбрать мобильный компьютер.

Адвокат по вопросам страхования жизни Марк Джонсон из Канзас Сити работает не так, как его коллеги. Каждый раз в зал суда он приносит портативный ПК и принтер. В ходе слушания дела он с молниеносной быстротой реагирует на происходящие события: находит нужный пункт в удаленной базе данных по законодательству или проецирует иллюстративные материалы с экрана компьютера для показа присяжным.

Немногие профессионалы сегодня используют технологию ПК так же интенсивно, как Джонсон, но, вероятно, скоро подобным образом будут работать многие. Мобильные компьютеры становятся все легче, производительнее и более приспособленными для связи. И все же самое приятное то, что цены на них постоянно снижаются, особенно на системы с монохромным дисплеем.

При выборе такого компьютера приходится учитывать множество характеристик: вес, размеры, продолжительность работы от батарей, объем памяти, средства связи, размеры клавиатуры, качество изображения, программное обеспечение, возможности расширения и цену. И все же первым шагом должно стать определение задач, которые вы планируете решать в дороге.

Если вы много путешествуете и хотите, чтобы ваша деятельность тем не менее была

хорошо организована, то необходим карманный компьютер. Подобные «крошки» весят всего несколько сотен грамм, месяцами работают от одного комплекта батарей, выполняя функции секретаря: отслеживают календарь встреч, ведут записи, производят вычисления и запоминают информацию о деловых контактах.

Если вы хотите иметь полноценный ПК минимального веса и размеров, подумайте о приобретении субблокнотного компьютера (subnotebook). Это легкие ПК-совместимые системы без НГМД, обмен данными и программами между которыми осуществляется по кабелю, через модем или PCMCIA-карты.

Правда,

PCMCIA-карты значительно дороже, чем гибкие диски, зато они обеспечивают большую универсальность, поскольку могут использоваться не только как средство хранения программ и данных, но и как средство подключения к системам связи.

Прежде чем принять окончательное решение о приобретении мобильного компьютера, попробуйте устройство ему небольшие испытания. Выясните, удобно ли ложатся пальцы на клавиатуру и не устают ли глаза от экрана без подсветки.

Если вы хотите иметь малогабаритное устройство с вычислительной мощностью настольного компьютера, рассмотрите вариант приобретения полнофункционального блокнотного ПК с монохромным или цветным дисплеем. Подобные устройства популярны среди путешествующих представителей торговых фирм, административных работников и писателей, так как по характеристикам почти не отличаются от настольных ПК, только стоят дороже. Если вы намереваетесь работать в дороге, оцените вес системы, качество изображения и размер клавиатуры.





Не забудьте обратить внимание на средства связи, которая всегда была одной из основных функций портативного компьютера. «Блокнотный ПК — это не вещь в себе; чтобы использовать все его возможности, нужно поддерживать связь с другими системами», — говорит Эндрю Уотсон, директор по маркетингу отделения портативных компьютеров корпорации Compaq.

Выбирайте блокнотный ПК с широкими возможностями подключения дополнительных

плат: модемов, факсов, сетевых адаптеров и адаптеров беспроводных ЛВС, позволяющих поддерживать связь в любое время и из любого места.

Если вы хотите приобрести блокнотный ПК взамен настольной машины, рассмотрите модели с установочной станцией (docking station), которая остается в офисе, а вместе с блокнотным ПК обеспечивает возможности применения полноразмерной клавиатуры, улучшенной графики, сетевых подсоединений и увеличения емкости жесткого диска.

ДЕСЯТЬ ВОПРОСОВ, НА КОТОРЫЕ СЛЕДУЕТ ОТВЕТИТЬ, ВЫБИРАЯ МОБИЛЬНЫЙ КОМПЬЮТЕР

1. Не обременителен ли вес устройства?
2. Используются ли в нем легкие щелочные аккумуляторы вместо более тяжелых никель-кадмиевых?
3. Достаточно ли велика продолжительность работы без перезарядки аккумуляторов и есть ли режим приостановки для экономии энергопотребления?
4. Удобна ли клавиатура для продолжительной работы?
5. Изображение четкое, ясное, легко читаемое при разных уровнях освещения?
6. Устраняет ли вас объем памяти системы?
7. Можно ли будет впоследствии увеличить емкость НЖМД и объем ОЗУ?
8. Имеются ли необходимые средства связи, в первую очередь скоростной модем, позволяющий посылать файлы в офис, получать данные и осуществлять дистанционное подключение к ЛВС?
9. Если вы покупаете карманную машину, попытайтесь оценить, сколько времени вы потратите на освоение встроенного в нее программного обеспечения.
10. Можно ли будет получить техническое обслуживание и помощь в дороге?



Благодаря установочной станции не нужно каждый раз, уезжая или возвращаясь в офис, переносить данные с системы на систему.

Портативные компьютеры стремительно развиваются и совершенствуются. Если вы всегда хотите использовать последние достижения техники, обратите внимание на машины, в которых имеется возможность обновления ключевых компонентов, например процессора, жесткого диска или дисплея.

В статье использованы фотографии, любезно предоставленные московским представительством корпорации Intel.

Мобильные компьютеры при восьми километрах в секунду

Филип Чен

Даже космический корабль, оснащенный мощными бортовыми компьютерами, уже не может обойтись без нескольких портативных ПК, способных решать сложнейшие технические задачи.

В мае 1991 г. на высоте более 320 км над Гавайскими островами астронавты Рик Хиб, Пьер Тьюо и Том Эйкерс в течение восьми с лишним часов находились в открытом космосе и совершили, казалось бы, невозможное — отремонтировали вышедший из строя спутник связи Intelsat весом 4063 кг. Это не только спасло «севший на мель» спутник стоимостью 131 млн. долл., но и ярко продемонстрировало потенциальные возможности современных мобильных компьютеров.

Когда астронавты покидали воздушный шлюз, чтобы начать свой отчаянный незапланированный выход в космос, пилот Кевин Чилтон использовал портативный компьютер для расчета расстояния от «Шаттла» до спутника. На экране он видел изображение «Шаттла» и путь, который ему предстояло пройти до встречи со спутником. Чилтон передавал

информацию командиру корабля Дэну Брэндстейну, который осторожно приближал «Шаттл» к неисправному спутнику.

Программа, которую использовал Чилтон, была создана астронавтом Риком Хибом, программистом-любителем, который написал несколько модулей, помогающих ему в работе.

Хотя Хиб создал свою программу для настольного компьютера, она была использована на «Шаттле», потому что портативные и настольные компьютеры совместимы: работают в одной и той же DOS и используют одно и то же семейство процессоров корпорации Intel.

Портативные ПК впервые отправились в космос в 1983 г. — для помощи экипажу в управлении космическим кораблем и проведении экспериментов. Теперь часто во время полета на борту «Шаттла» практи-

чески столько же мобильных компьютеров, сколько и астронавтов. Мобильные ПК дополняют бортовые компьютеры и обеспечивают астронавтов собственной информацией, так что последним нет необходимости полагаться все время на пункт управления полетом.

Сегодня мобильные компьютеры NASA, называемые компьютерами общей поддержки (Payload and General Support Computer — PGSC), — это серийные компьютеры корпорации Grid Computer System модели 1530 со стандартным процессором 386DX, 8-Мбайт ОЗУ и 40-Мбайт жестким диском. NASA считает портативные ПК настолько разносторонними, что каждый раз на борту «Шаттла» их оказывается несколько штук.

Во время полета летом 1992 г. космическая лаборатория микрогравитации (Spacelab Microgravity Laboratory) имела на борту шесть портативных ПК. Они использовались в качестве дисплейных терминалов компьютеров космической лаборатории, для управления экспериментами и обмена данными с Землей.

Во время другого полета, более всего известного неудачной попыткой запустить экспериментальный генератор электрического тока, астронавт-ученый Франклин Чанг-Диас использовал два портативных ПК корпорации Grid для сбора данных с огромного количества научных приборов. «Это новая область применения компьютеров — первый знак того, что в будущем на «Шаттле» основные операции при проведении научных исследований и экспериментов будут проходить непосредственно под управлением портативного компьютера», — сказал Чанг-Диас.

Пионеры информационной эры

Брайан Хастингс

Наступает время, когда компьютер становится по-настоящему мобильным и общение с ним превращается в праздник, который всегда с тобой.

Когда последние лучи солнца покидают долину Ламар реки Йеллоустон, из своего логова выходит койот, чтобы найти пищу для детенышей. Где-то пробирается к ручью утолить жажду медведь гризли. Жизнь этих постоянных обитателей долины протекает так же, как и тогда, в начале девятнадцатого века, когда через нее прошел первый охотник.

Теперь человек явился в долину для сохранения ее обитателей и принес вместо ловушек мобильный компьютер. Роберт Крэбтри (профессор университета шт. Монтана) и группа исследователей проводят всестороннее изучение диких животных и растительности национального парка. С помощью карманных компьютеров HP 95LX компании Hewlett-Packard ученые наблюдают за койотами и фиксируют их охотничьи тропы. Рядом с норами и со свежими лосинами тушами установлены дополнительные радиофицированные карманные компьютеры, которые принимают и записывают информацию о ноч-

ных передвижениях койотов, снабженных ошейниками с передающими устройствами. По склонам холмов ходят «самодельные ученые» профессора Крэбтри — добровольцы, которые с помощью сложной программы идентификации для карманного ПК определяют растения и записывают результаты.

«Карманные компьютеры позволяют нам нанимать непрофессиональных работников для сбора данных, что обычно требовало привлечения группы ученых, — говорит Крэбтри. — При этом резко сокращаются затраты на проводимое нами исследование возможностей гармоничного сосуществования людей и обитателей парка».

Преследуя те же цели, его коллега Ричард Найт создал систему определения местоположения медведей гризли с помощью спутника. Прикрепив приемник сигнала глобальной спутниковой системы местопредопределения (GPS — Global Positioning System) к блокнотному ПК Zeos, лесничий может, облетая территорию парка,

фиксировать перемещения снабженных передатчиками медведей. На этом ПК исполняется программа, которая переводит переданные через спутник сигналы в координаты. «Система GPS позволит нам преградить путь медведю, прежде чем он слишком близко подойдет к местам расположения лагерей», — говорит Найт.

До того как появиться в этой глуши, на северо-западе штата Вайоминг, мобильные компьютеры были освоены корпорациями, такими, как Citibank, AT&T и фармацевтическая фирма CIBA-Geigy, оснастившими тысячи своих торговых представителей и сотрудников выездных сервисных служб портативными и блокнотными ПК. Для служащих, которые много времени проводят в пути, как, например, внутренние аудиторы (ревизоры) фирмы Citibank, портативный компьютер — главный инструмент. «Аудиторы могут провести несколько недель в филиале банка, чтобы определить соответствие деятельности банка правительственной и внутренней политике, — говорит Роберт Кейлипитс, глава аудиторской службы фирмы Citibank. — Они при этом могут взять с собой свой офис, что и делают, беря портативный ПК».

«Виртуальный» офис

Идея помещения своего офиса в кейс послужила основой для развертывания еще одного смелого исследования в области мобильных компьютеров — программы «виртуального» офиса корпорации AT&T. В начале 1991 г. эта фирма выдала своим торговым представителям на коммерческих рынках блокнотные ПК Safari, оборудованные факс-модемом, портативные принтеры корпорации Canon, платы автоматического



Компьютерная техника везде и всегда: Роберт Крэбтри работает в Йеллоустонском национальном парке, где карманные компьютеры помогают сохранять природу.

набора телефонных номеров и (в некоторых случаях) сотовые телефоны. Целью было, по словам руководителя программы Дуга Айдлмэна, позволить сотрудникам для повышения эффективности их работы меньше времени проводить в офисе и больше — среди клиентов. Освободившись таким образом от существующих стереотипов организации, АТ&Т начала сокращать количество офисов и уменьшать занимаемые ими площади. Теперь эти представители АТ&Т посещают либо местные офисы, обеспечивающие основные услуги, либо более полно оснащенные центральные офисы, которые обеспечивают техническую и секре-

тарскую поддержку, а также предоставляют комнаты для встреч. «Мы сократили на 25% наши расходы на недвижимость, экономя миллионы долларов ежегодно», — говорит Айдлмэн.

Домашний офис

Айдлмэн признает, что перенос рабочего места из офиса куда-либо еще не обходится без проблем. Работники теряют некоторые традиционные символы защищенности и товарищества: ежедневное посещение офиса, семейное фото на рабочем столе, утренние перерывы у автомата с охлажденной водой, так что, естественно, были

опасения. Некоторые менеджеры также сомневались в возможности управления людьми, которые теперь приходят в офис дважды в месяц или еще реже. Айдлмэн заявляет, что как управляющие, так и рядовые сотрудники после нескольких месяцев приспособились к режиму «виртуального» офиса и что объем продаж стал возрастать в первый же год. «Управляющие и их представители поддерживают связь практически каждый день по телефону и с помощью электронной почты, — сообщает Айдлмэн, — и они находят другие пути поддерживать контакты со своими коллегами, например, встречи раз в неделю в ресторане».

Некоторые торговые представители вполне довольны тем, что они больше времени проводят со своими семьями, а не в офисе. Дэйв Олсон из Джексона (шт. Миссури) живет в трех часах езды от ближайшего офиса АТ&Т, поэтому он оборудовал офис у себя в подвале. «Теперь я могу несколько раз в неделю посидеть за ленчем со своей женой и двухлетними дочерьми-близнецами», — говорит Олсон. Но он предупреждает о необходимости не смешивать дом и рабочее место.

Айдлмэн дает два совета компаниям, желающим подобным образом автоматизировать свои коммерческие операции: точно настраивать программу на конкретные задачи группы и учиться, учиться, учиться. ♦

Справочник покупателя

Анита Амиррежвани

Вы хотите работать в дороге? Тогда Вам нужен мобильный компьютер. В данном обзоре мы представляем новейшие изделия, уже имеющиеся в продаже. Речь идет о разных видах компьютеров, начиная с персонального электронного секретаря стоимостью 300 долл. до цветного портативного компьютера ценой 5500 долл. Итак, что же купить?

КАРМАННЫЕ КОМПЬЮТЕРЫ

Самый легкий компьютер для работы в дороге — карманный. Он весит примерно 500 г и позволяет всегда иметь с собой график запланированных встреч и базу данных с именами и адресами, выполнять вычисления с помощью электронной таблицы и даже пользоваться прикладными пакетами для планирования инвестиций и ведения торговых контрактов. В карманных компьютерах не бывает роскошных клавиатур и больших экранов, а объем памяти обычно относительно мал — не более 1 Мбайт.

APPLE NEWTON

Это не детский альбом для рисования, а записная книжка, которой могут пользоваться и дети, и взрослые. В ней нет клавиатуры — вместо нее используется электронное «перо», позволяющее «писать» и рисовать прямо на экране. Специальная программа преобразует рукописные буквы в печатные, затем текст распознается и соответствующим образом обрабатывается. Если написать* «Позвони Эми Рассел», Newton выведет на экран номер телефона и всю остальную информацию, имеющуюся о Рассел. Ожидается, что это изделие появится в продаже в начале 1993 г., будет весить менее 500 г и стоить менее 1000 долл.

Apple Computer, Inc.
Представительство в Москве
Тел. (095) 229-11-36

ATARI PORTFOLIO

При цене 300 долл. и весе 450 г это устройство позволяет разместить целый офис в кармане. Оно включает



Apple Newton

DataStor 64

HP 95LX

*Распознавание рукописной кириллицы пока не предусмотрено. — (Прим. ред.)



Atari Portfolio

табличный процессор, электронный секретарь, телефонный справочник и при этом обеспечивает обмен данными с настольными ПК. Объем памяти может достигать 512 Кбайт. Atari Corp. 1-408/745-2098

DATAStor 64

Персональный помощник DataStor 64 стоит 150 долл., хранит до 2000 имен и номеров телефонов и имеет встроенные калькулятор, электронный секретарь, календарь и систему ведения банковских счетов. Его размеры 114 × 70 × 19 мм, вес 170 г. Vi-Tel Electronics 1-800/443-0594

DIGITAL BOOK SYSTEM

Хотите быстро найти нужное слово, рецензию на фильм или медицинскую справку? Вставьте специальную карточку с информацией в одно из двух гнезд этого компьютера и получите доступ к 45 Мбайт данных. В комплект поставки входит словарь Merriam-Webster's Dictionary Plus и справочник Word Games. Весь комплект весит 127 г (включая батареи) и стоит 200 долл. Franklin Electronic Publishers 1-800/762-5382

Digital Book System

HP 95LX

Этот компьютер весит всего 312 г и среди имеющихся на рынке персональных помощников является одним из самых разносторонних. Уплатив 550 долл., вы получите устройство со встроенной электронной таблицей 1-2-3 корпорации Lotus Development, телефонным справочником, калькулятором, электронным секретарем, текстовым редактором и коммуникационным пакетом. Но это еще не все: вы можете при необходимости расширить объем памяти, установить новое программное обеспечение и даже подключить беспроводной модем.

Hewlett-Packard

Представительство в Москве
Тел. (095) 181-80-02

Sharp Wizard OZ-9600

Клавиатура этого компьютера выполняет только вспомогательную функцию — управлять его работой можно касаясь экрана пальцем. Как и в системе Newton, с помощью специального «пера» можно писать заметки или создавать рисунки. Компьютер Wizard выполняет все присущие персональному секретарю функции, поддерживая, в частности, список текущих дел, в котором незавершенные задачи автоматически переносятся на следующий день. Ожидается, что стоит устройство Wizard будет менее 500 долл., а в продаже появится в начале 1993 г. Sharp Electronics 1-800/321-8877

Sharp Wizard OZ-9600

Монохромные блокнотные ПК

В настоящее время на рынке имеются десятки блокнотных ПК с монохромными дисплеями. Самые дорогие и высокопроизводительные системы оснащаются микропроцессором 486 корпорации Intel, НЖМД емкостью 120 Мбайт, НГМД, ОЗУ большого объема (достаточного для работы с любыми графическими пакетами) и дисплеями с разрешением SVGA, не уступающими хорошим мониторам настольных систем.

AST POWEREXEC

Хотите иметь наиболее гибкую систему? Если вы устали от монохромного экрана, подсоедините факультативный цветной дисплей. Вставьте в разъем PCMCIA плату с факс-модемом, дополнительной памятью или даже миниатюрным жестким диском. Модель PowerExec весит 2,7 кг, стоит 2395 долл., а время работы без перезарядки аккумуляторов составляет от 4 до 6 час. AST Research, Inc. 1-800/876-4278

COMPAQ CONTURA 3/25 MODEL 60W

Этот компьютер весит 2,7 кг, стоит 2300 долл. и обладает самыми разнообразными функциональными возможностями. Объем ОЗУ и емкость НЖМД достаточны для нормальной работы поставляемых с ним системы Windows и пакета PFS:WindowWorks. Для увеличения времени работы без перезарядки аккумуляторов в модели Contura используется несколько способов управления энергопотреблением.

Compaq Computer Corp.

Дистрибутор в Москве — Computerland
Тел. (095) 243-35-53

DELL DIMENSION NL25

Высокая производительность при невысокой стоимости — так корпорация Dell заманивает экономных покупателей, предлагая блокнотный компьютер Dimension NL25. Кроме малой цены (1649 долл.), он имеет множество других достоинств: производительность процессора 386, четкое изображение, 32 оттенка серого, расширяемая долговременная и оперативная память, вес 2,7 кг. Если вы не в пути, можно подключить к нему модем, манипулятор «мышь» и даже полноразмерный монитор.

Dell Computer Corp.

Дистрибутор — ELCO Technology
Тел. (095) 138-18-05

EPSON NB-SL/25

Компьютер Epson предоставляет пользователю дополнительные удобства. В модели SL/25 используются





Compaq Contura 3/25 Model 60W



Dell Dimension NL25



NCR System 3170



Z-Note 325L Model 60

сменные кассеты с НЖМД различной емкости. Перенос данных на настольную систему предельно прост: вынутый из блокнотного ПК НЖМД вставляется в специальный адаптер, подключенный к настольному ПК. Модель Epson NB-SL/25 весит 2,95 кг и стоит 2299 долл.

Epson America 1-800/289-3776

GRID CONVERTIBLE

Уникальное сочетание многообещающей технологии рукописного ввода и удобства обычной клавиатуры. Компьютер Convertible, весящий 2,5 кг, можно использовать как планшет, делая записи и рисуя электронным «пером». Если нужно работать с текстом, можно приподнять экран — откроется спрятанная под ним полноразмерная клавиатура. «Перо» работает и одновременно с клавиатурой, заменяя «мышь». Цена модели Convertible чуть ниже 3000 долл.

Grid Systems Corp. 1-800/222-4743

NCR SYSTEM 3170

Компьютер корпорации NCR — это странствующий рыцарь передачи данных. Он стоит 3420 долл. и поставляется со скоростным факс-модемом. Подключите к нему сотовый телефон — и вы сможете передавать и принимать данные, забравшись в самую глушь. Компьютер System 3170 очень легкий (2,3 кг), правда, не имеет встроенного НЖМД.

NCR Corp. 1-800/225-5627

POWERBOOK 145

Блокнотный ПК PowerBook 145 — это полнофункциональный компьютер, совместимый с ПК Macintosh. Для управления курсором в нем предусмотрен шаровой манипулятор, а клавиатура заглублена, поэтому во время набора текста кисти рук могут отдыхать. Имеется динамик и микрофон. Аккумуляторов хватает на три часа работы. Базовая модель с жестким диском емкостью 40 Мбайт стоит менее 2400 долл.

Apple Computer

Представительство в Москве

Тел. (095) 229-11-36

Z-NOTE 325L MODEL 60

Если вы часто бываете в нескольких офисах и хотите постоянно иметь доступ к информации, имеющейся в различных ЛВС, вы по достоинству оцените компьютер Z-Note (ценой 3599 долл.). Он поставляется с установленным программным обеспечением трех наиболее популярных сетевых систем (VINES, LAN Manager и NetWare). Большой монохромный экран (25 см по диагонали) можно заменить цветным.

Zenith Data Systems

Дистрибутор в Москве — Computerland

Тел. (095) 243-35-53

СУББЛОКНОТНЫЕ ПК

При весе не более 1,6 кг и неправдоподобно малых размерах (127 × 229 мм), субблокнотные ПК тем не менее обладают многими возможностями полноразмерных портативных компьютеров. В субблокнотных ПК нет накопителей на гибких дисках, но их обычно можно приобрести как внешнее факультативное устройство. Обмениваться информацией с другими ПК можно путем пересылки данных по кабелю или телефонной линии. Если вам нужна высокая производительность, но не нужен лишний вес, субблокнотный ПК — это именно то, что вам нужно.

DELL 320 SLi

Модель 320 SLi стоит всего 1969 долл, весит 1,6 кг и все же имеет микропроцессор 386, а также достаточно емкий НЖМД (60, 80 или 120 Мбайт) и большой объем ОЗУ. Клавиатура удобная, а экран достаточно большой (24 см).

Dell Computer Corp.

Дистрибутор — ELCO Technology

Тел. (095) 138-18-05

GATEWAY HANDBOOK

Компьютер корпорации Gateway 2000 весит 1,2 кг, а его ширина всего 230 мм. По быстродействию и возможностям расширения он уступает изделию корпорации Dell, однако может работать практически с любым программным обеспечением для настольных ПК (хотя с графическими пакетами, например Windows, могут возникнуть сложности). Размер экрана меньше, чем у компьютера Dell (19 см по диагонали). Если вы цените портативность больше, чем производительность, обратите внимание на модель HandBook, которая стоит всего 1295 долл.

Gateway 2000 1-800/523-2000

ZEOS POCKET PC

По габаритным размерам модель Pocket PC занимает место между карманным компьютером и субблокнотным ПК. Она весит всего 0,59 кг, однако имеет довольно большую клавиатуру (длина 230 мм). В комплект поставки компьютера входит пакет Microsoft Works, который включает в себя редактор текста, электронную таблицу, базу данных, коммуникационную программу и электронный секретарь. Модель Pocket PC не имеет жесткого диска, однако оснащена двумя гнездами PCMCIA, к которым можно подключать карты, заменяющие жесткие и гибкие диски.

Zeos Int'l, Ltd. 1-800/423-5891



Dell 320 SLi

Gateway HandBook

Zeos Pocket PC

ЦВЕТНЫЕ БЛОКНОТНЫЕ ПК

Наконец-то можно путешествовать с ПК с цветным экраном! За 4000 долл. сегодня можно приобрести портативную систему весом от 3 до 4 кг с хорошим цветным экраном и всеми остальными характеристиками обычного ПК. Если же вы не стеснены в средствах и не обделены силой, можете приобрести более тяжелый (до 9 кг) цветной портативный компьютер, экран которого отображает 256 цветов и обеспечивает хорошую яркость и разрешение адаптера SVGA.

COMPAQ LTE LITE/25C

При размерах с пачку писчей бумаги толщиной 38 мм и цене 4999 долл. этот компьютер имеет все возможности быстрой настольной системы с процессором 386, а весит всего 2,95 кг. В нем имеется встроенный шаровой манипулятор и превосходный экран на основе активной матрицы, цвета на котором могут неискушенного человека привести в восторг. Благодаря хорошей системе экономии энергопотребления время работы без подзарядки аккумуляторов составляет 3—4 часа.

Compaq Computer Corp.

Дистрибутор в Москве — Computerland
Тел. (095) 243-35-53

SHARP PC-6881

Этот компьютер легок как перышко — весит всего 2,7 кг. Он стоит 4799 долл., имеет процессор 386SL и емкий жесткий диск (80 Мбайт). Имеется встроенный шаровой манипулятор. Экран отображает 256 цветов, причем изображение яркое и четкое. Однако продолжительность работы от аккумуляторов невелика — от 40 до 90 мин. Можно подключать внешний блок аккумуляторов.

Sharp Electronics 1-800/321-8877

TOSHIBA 4400SXC

Как и модель Compaq LTE, компьютер компании Toshiba обеспечивает яркие и насыщенные цвета, однако имеет большой вес (3,5 кг). Благодаря процессору 486 и дополнительной видеопамяти он быстр как молния и его цена (5499 долл.) вполне оправдана. Экран на основе активной матрицы имеет размер 22 см по диагонали и отображает 256 цветов. Время работы без подзарядки аккумуляторов около 3 часов.

Toshiba America Information Systems
1-800/344-3445

ОБ АВТОРЕ

Анита Амирежвани — внештатный редактор журнала PC World.

Анатомия карманного и блокнотного компьютеров

Стив Басс

КАРМАННЫЙ КОМПЬЮТЕР

В карманном компьютере Hewlett-Packard 95LX высокая производительность самым парадоксальным образом сочетается с малым весом. Он весит всего лишь 311 г, однако среди машин своего класса является одним из самых универсальных. Среди множества достоинств этой модели следует особо отметить встроенные электронную таблицу 1-2-3 корпорации Lotus Development и средства получения электронных сообщений. Розничная цена — 550 долл.

1 ДИСПЛЕЙ

Черно-белый жидкокристаллический дисплей размерами 11,4 × 6,4 см отображает 16 строк по 40 символов. Это типичные параметры дисплея карманной машины.

2 КЛАВИАТУРА

Если вы печатаете вслепую, будьте осторожны: клавиатура компьютера 95LX является кукольной версией клавиатуры пишущей машинки и имеет специализированные клавиши для вызова встроенных программ.

3 ПЛАТЫ РАСШИРЕНИЯ ПАМЯТИ

Сменные платы памяти, имея размеры кредитной карточки, вмещают до 4 Мбайт программ и данных — объем, достаточный для записи текста книги «Война и мир».

4 БАТАРЕИ

В большинстве карманных компьютеров используются батареи типа AA. Модель 95LX не является исключением. Одного комплекта батарей хватает примерно на два месяца.

5 ВСТРОЕННЫЕ ПРИКЛАДНЫЕ ПРОГРАММЫ

Многие карманные компьютеры

предназначены для решения одной единственной задачи. В модель 95LX встроена электронная таблица 1-2-3, финансовый калькулятор, электронный секретарь, телефонный справочник, записная книжка и коммуникационная программа для получения электронных сообщений.

6 ВСТРОЕННЫЙ ДИНАМИК

Даже в столь малых компьютерах бывают динамики, сигнализирующие о выполняемых программами действиях.

7 ПРОЦЕССОР

«Мозгом» 95LX является процессор, аналогичный процессору IBM PC XT, т.е.

соответствующий последним достижениям настольных компьютеров 1983 г.

8 ПАМЯТЬ

Компьютер 95LX имеет оперативную память объемом 1 Мбайт, что достаточно для выполнения большинства популярных программ DOS.

9 СЕТЕВОЙ БЛОК ПИТАНИЯ

Используя блок питания для работы от сети переменного тока, вы можете сэкономить ресурс батарей.

10 ПОРТ ВНЕШНИХ УСТРОЙСТВ

С помощью специальных кабелей к разъему этого порта можно подсоединять принтер, модем, другие ПК и прочие устройства.

11 ИНФРАКРАСНЫЙ ПОРТ

Устройство связи на инфракрасных лучах компьютера 95LX позволяет без всяких соединительных кабелей обмениваться информацией с другими компьютерами, а также печатать документы на лазерных принтерах компании Hewlett-Packard.

12 РЕЗЕРВНАЯ БАТАРЕЯ

Резервная литиевая батарея напряжением 3 В обеспечит сохранность ваших данных во всех случаях, даже если удалить основные батареи.



БЛОКНОТНЫЙ КОМПЬЮТЕР

Модель PowerExes фирмы AST весит всего 2,5 кг и имеет размеры 212 × 279 × 46 мм, что примерно соответствует размерам папки для бумаг. Вес блокнотных ПК обычно составляет от 2 до 3,5 кг, причем в них используются те же процессоры, что и в настольных ПК. Некоторые блокнотные ПК, в том числе PowerExes, имеют цветные экраны. Во многие встроен факс-модем, позволяющий посылать и получать документы от факс-аппаратов и других компьютеров. Розничная цена — 2395 долл.

1 ДИСПЛЕЙ

Дисплей большинства блокнотных ПК черно-белые (монохромные), однако в последнее время появляется все больше цветных. Экран компьютера фирмы AST имеет типичный для блокнотных ПК размер — 24 см по диагонали. Он отображает до 256 цветов одновременно, причем по качеству изображения вполне может соперничать с мониторами настольных компьютеров.

2 КЛАВИАТУРА

Клавиатура этого «блокнота» похожа на клавиатуру пишущей машинки, однако имеет дополнительные клавиши для специальных команд.

3 БАТАРЕИ

В зависимости от режима использования компьютера и режимов работы системы управления энергопотреблением батареи обеспечивают непрерывную работу компьютера PowerExes от трех до шести часов. Перезарядить их можно всего за 90 минут.

4 СИСТЕМА УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ

Блокнотный ПК PowerExes следит за действиями пользователя и временно отключает неиспользуемые компоненты компьютера для экономии ресурса аккумуляторов.

5 ДИНАМИК

В блокнотных ПК динамик используется для выдачи сигналов о разряде аккумуляторов и о выполняемых программами действиях.

6 НГМД

Жесткий диск компьютера PowerExes имеет емкость 160 Мбайт, что

эквивалентно емкости более 100 гибких дисков или 80 тыс. страниц машинописного текста.

7 ПРОЦЕССОР

В модели PowerExes используется процессор 386SL корпорации Intel, предназначенный для систем с пониженным энергопотреблением.

8 ПАМЯТЬ

ОЗУ необходимо процессору во время работы для временного хранения программ и данных. Обычно блокнотные ПК поставляются с ОЗУ емкостью 2 Мбайт и имеют возможность его расширения.

9 НГМД

В блокнотном ПК PowerExes используются стандартные диски емкостью 1,44 Мбайт.

10 РАЗЪЕМ РСМCIA

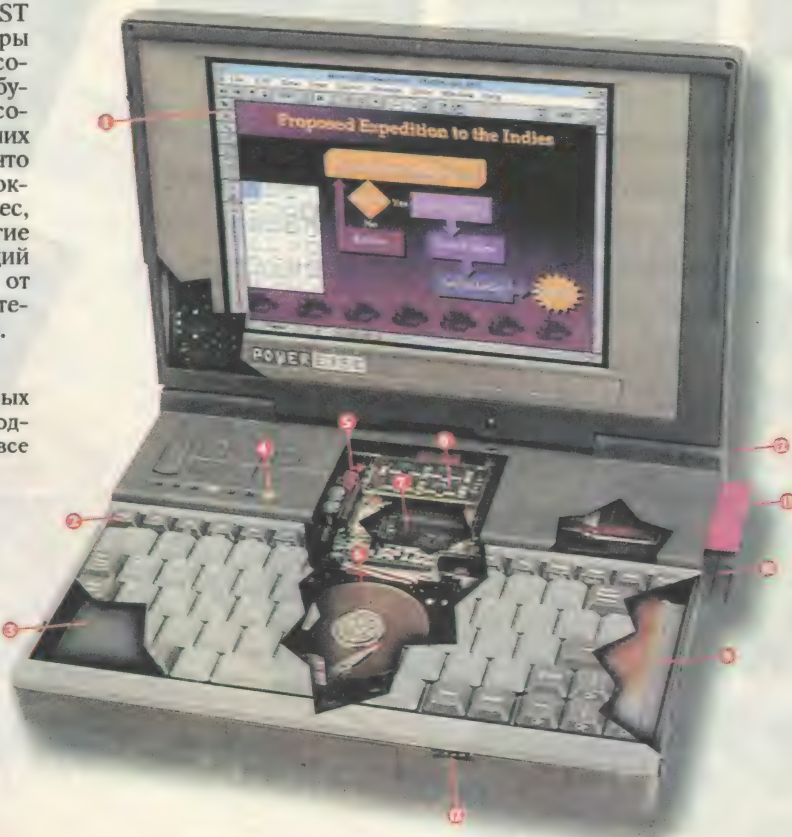
Наличие этого разъема позволяет подключать карты памяти, карты модемов и контроллеров ЛВС.

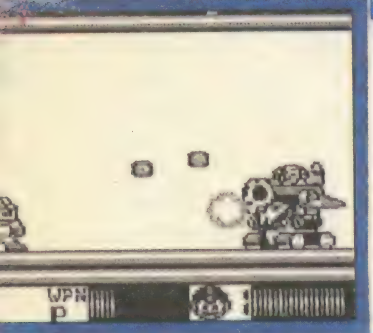
11 МОДЕМ

Вставив в разъем РСМCIA карту 2400-бит/с модема, вы сможете по линиям телефонной связи посылать и получать электронные сообщения со скоростью более 2400 слов в минуту.

12 ПОРТЫ

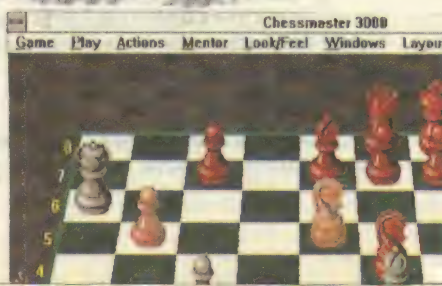
Разъемы портов позволяют подключать к блокнотному ПК манипулятор «мышь», принтер, модем и другие устройства. ♦





Показывать, чтобы увидели

М.В. Донской



Начну издалека. Из Канады. Там на одной конференции я слышал доклад замечательного полемиста Парнаса (знаменитого статьей «Нужен ли инженерам искусственный интеллект?») на тему «Как отлаживать программы, которые никогда не работают». Название забавное, но первый же пример такой программы показал всю серьезность предмета. Это была программа для отработки аварийной ситуации на атомной электростанции. Ясно, что ждать отладки до реальной аварии нельзя. В этой теме много интересных аспектов, но один был просто замечательным.

Парнас предложил всем слушателям прикинуть, как написать программу, которая реализовала бы проверку условия: «В течение пяти минут температура котла была выше 200 градусов по Цельсию». Оказалось, что существует **ДЕСЯТЬ** различных корректных способов запрограммировать такую проверку, из которых только один правилен для этого конкретного приложения. (Мой вариант — минимум температуры котла в течение пяти минут был выше 200 градусов — абсолютно неверен.) Проблема в том, что, не зная физического смысла этого условия, нельзя выбрать правильную интерпретацию.

Но ведь вся математика, в том числе и вычислительная, абстрагируется от деталей. Какие из них существенны, а какие нет? И тут мы подходим к компьютерной анимации. Рассмотрим самую простую задачу по анимации — нарисовать падающий со стола резиновый мячик.

На этом простейшем примере уже можно увидеть совершенно удивительную для программистов и математиков вещь, которая для аниматоров настолько очевидна, что и обсуждения с их точки зрения не заслуживает. Я бы, вспомнив

школьную физику, рассчитал траекторию падения мячика и нарисовал все точки этой траектории с шагом 1/24 секунды. («Элементарно, Ватсон!»)

Элементарно, но неверно! Все аниматоры знают, что если среди нарисованных кадров не будет кадра, в котором мячик находится непосредственно на полу, то зритель увидит странную картину: мячик, отскакивающий от невидимой преграды недалеко от пола.

В чем тут дело? Просто при просмотре анимационного фильма присутствуют два физических процесса, а не один. Наряду с физикой движения есть физика восприятия. И последнюю очень важно учитывать для правильного восприятия движения. То есть, если надо абстрагироваться от чего-либо, то уж скорее от математически правильных формул движения, чем от психологически необходимых формул восприятия.

Говоря точнее, компьютерная анимация может пониматься как проекция физически непрерывного движения на дискретное пространство (эту проблему все программисты, имеющие опыт в компьютерной графике, чувствуют хорошо) и дискретное время (а тут уже место парадоксам Парнаса).

Художники-аниматоры свободно манипулируют временем, более того, для них это — одно из главных средств эмоционального воздействия на зрителя (см. статью А. Орлова «Анимация в компьютерных играх» на с. 33). Способ движения персонажа задает его характер, физические несообразности процессов понимаются как интонационные акценты. Программисты же цепляются за «правильность» физических процессов...

Тут ситуация близка к знаменитому чапековскому петуху: «Мое дело прокукарекать, а утро хоть не наступай». Основ-

ная забота программистов — правильно нарисовать, а уж правильно ли это смотрится — не их дело. Я разговаривал с программистом, который понимал, что в его игре движения не выглядят естественными, но, проверив правильность формул и кода и убедившись, что ошибок в них нет, он смирился, считая, что такова судьба. Мой совет чуть-чуть исправить скорость движения, чтобы ключевые моменты движения все-таки были визуализированы, вызвал у него ужас. Правильность физической модели была, с его точки зрения, священна.

Это отражает особенность нашего научного образования, априори предполагающего существование абсолютной истины. Если же мы не можем познать эту истину, это наша проблема.

С таким подходом связано и представление Андрея Родионова о могуществе создателя новых программно-игровых миров (см. статью «Своя игра» на с. 42). Его больше интересует гармоническое саморазвитие построенного им мира, чем гармоничное восприятие этого мира наблюдателем. В его философии явно не хватает Олимпа — собрания богов, только один из которых является создателем мира, а остальные — наблюдают за этим миром.

Однако есть и другая точка зрения — та, что мир существует постольку, поскольку мы его воспринимаем. Приложение этой высокой философской посылки к компьютерной анимации означает, что нужно исходить из того, что человек должен увидеть правильный процесс, а будет ли правильной его математическая интерпретация — дело второе.

Кстати, аналогичная проблема блестяще решается в живописи на протяжении многих веков. Существует теория композиции, которую можно све-

сти к тому, как нарисовать неправильно, но так, чтобы зритель увидел правильно. Глядя на иконы, я иногда свысока думал, что вот, мол, нарисовано неправильно, а впечатление производит сильное. А вот если бы еще и правильно нарисовать! И мне было невдомек, что эмоциональный эффект напрямую связан с «нефизичностью» иконописи, что художники специально, хотя, может, и подсознательно, искажали картину так, чтобы взгляд сосредоточивался на том единственном месте, при фокусировке на котором не раздражали бы искажения.

При переходе от статической картины к анимационной проблема усложняется. Если в статической картине зритель может постепенно рассматривать все ее части, переводя взгляд от одной к другой (хотя искусство художника вынуждает его все-таки постоянно возвращаться к композиционному центру), то в движущейся картине все рассмотреть не успеет никто и никогда. И тщательное вырисовывание деталей, находящихся на периферии внимания зрителя, никакого смысла не имеет. Более того, оно может оказаться вредным, поскольку основная роль периферийного зрения, по-моему, и состоит в том, чтобы задать систему координат для рассматривания объектов, на которых сосредоточено сознательное внимание. Излишнее привлечение внимания к периферии может разрушить композицию и тем самым естественность восприятия.

В анимационной культуре существует понимание того, как минимальными изобразительными средствами добиваться тех эффектов в восприятии зрителя, на которые рассчитывает автор фильма. Существует своего рода инструментальный ящик, откуда со-

здатели анимационного кино выбирают необходимые им инструменты. И, конечно, начинающие аниматоры учатся у мэтров пользованию этими приемами. Интересно, много ли программистов, работающих в области компьютерной графики и особенно анимации, знакомы с ними?

В этой связи мне хотелось бы упомянуть инициативу Аниматографического центра ПИЛОТ, организовавшего цикл лекций по анимации для программистов. Конечно, программисты, как и все остальные, хорошо разбираются в кино, но знание секретов мастерства анимации, поверьте, может показать совершенно новые подходы и технические приемы. А мне эти лекции помогли в корне изменить отношение к изобразительным искусствам. Основное, что я понял, — манипулировать надо не объектом на экране, а зрительским восприятием. И, что очень важно, это реализуется совершенно разными способами.

Центр ПИЛОТ давно сотрудничает с программистами и кроме лекций имеет на своем счету организацию очень интересных семинаров по компьютерной анимации, в том числе в рамках конференции ГРАФИКОН. И спонсоры ему не помешали бы.

Есть уже и обратное влияние — влияние новейших компьютерных технологий на анимацию. Одной из самых трудоемких и неблагодарных операций при работе над анимационным фильмом является фазовка — рисование различных фаз движения персонажей от одного критического кадра (он называется у профессионалов компоновкой — это кадр, который необходим для правильного восприятия движения) к другому. В сущности, задача фазовки — интерполяция между компоновками.

Эта же задача интерполяции была поставлена и удачно решена как часть решения проблемы упаковки видеоизображений. В новом стандарте упаковки видеоизображений MPEG (Международной группы по видеостандартам, являющейся частью ISO — Международной организации по стандартам) требуется сохранение лишь некоторых базовых кадров (3—5 базовых кадров в секунду) и информации, необходимой для экстраполяции еще по одному кадру из каждого базового. Остальные кадры, необходимые для получения «видеокачества» — 30 кадров в секунду, получают прямой интерполяцией между базовыми и экстраполированными кадрами. То есть, как бы происходит фазовка между этими кадрами. (Стандарт MPEG — это развитие стандарта JPEG, подробно о котором написано в журнале «Мир ПК», № 4/92.)

Важность применения устройств, реализующих распаковочный алгоритм MPEG, позволяет надеяться на широкое их распространение и, следовательно, умеренную цену. А это, в свою очередь, позволит аниматорам использовать их в своей работе, если и не в автоматическом режиме, то по крайней мере как начальное приближение, позволяющее существенно уменьшить объем технической работы.

Еще одна польза, как говорил киплинговский Питон Слопенку, для аниматоров от компьютерной графики — это учет взаимодействия, создание фильмов, где зритель является активной стороной. Тут опыт программистов, особенно авторов компьютерных игр, может оказаться неоценимым. Но это — тема отдельного разговора.

ОБ АВТОРЕ

Михаил Владимирович Донской — канд. физ.-мат. наук, заведующий лабораторией ИСА РАН; e-mail: donskey@child.msk.su.

Анимация

в компьютерных играх

Алексей Орлов

Так ли часто встречаются «живые» компьютерные игры? Как создать такую игру?

Использование принципов диснеевской анимации поможет обеспечить комфортные условия восприятия экранных образов.

Современные компьютерные игры все шире используют традиционную анимацию. Однако на сегодняшний день анимационная эстетика компьютерной игры имеет ряд специфических качеств, отличающих ее от обычной анимации, — как авторской, так и коммерческой.

Анимация компьютерных игр, пожалуй, ближе к так называемой редуцированной анимации, чем к полной, или диснеевской.

Отличия между полнотой и редуцированностью можно пояснить на простом примере. Например, для того чтобы показать поворот персонажа из положения в фас в положение в профиль, у Диснея используется восемь-десять промежуточных фаз движения. В редуцированной анимации ограничиваются одной-тремя фазами. Или, скажем, при движении все тело диснеевского персонажа слегка изменяется по длине — то укорачивается, то удлиняется. Кроме того, центр тяжести тела перемещается по синусоиде. В редуцированной анимации движение персонажа сводится к простому сдвигу его корпуса в направлении движения. Такая редукция сильно со-

кращает объем работы аниматора и резко удешевляет производство.

В целом диснеевская анимация создает достаточно полную иллюзию достоверности движения в кадре, будь то качание ветки дерева при порыве ветра или замах руки персонажа перед броском камня. Не случайно Дисней обучал своих аниматоров, показывая им кинограммы (отдельные фазы движения) человека, животных, птиц и т.д.

Семиминутный фильм у Диснея состоял из 10—15 тысяч рисунков. Фильм, выполненный в редуцированной технике, требует в 5—6 раз меньшего количества рисунков.

Редуцированная анимация обостряет ощущение условности персонажа: его внешности, жестикуляции, походки, мимики и т.д. Это, как правило, препятствует отождествлению зрителя с персонажем. Зритель в этом случае перестает эмоционально сопереживать персонажу и достаточно отстраненно и холодно следит за интригой.

Анимация в компьютерных играх в целом близка к описанной выше редукции. Программисты практически не используют диснеевскую эстетику и ее законы и чисто стихийно при-

ходят к принципам редуцированной анимации с ее подчеркнутой условностью и схематизмом.

Повышению степени условности персонажа в компьютерных играх способствует и часто употребляемый прием «растворения» персонажа в момент гибели и его последующее появление из ниоткуда. На наших глазах как бы происходит дематериализация персонажа и сразу вслед за ней — новая материализация. Частое повторение таких моментов в ходе игры приводит к их восприятию как нормы бытия персонажа. При этом персонаж (к тому же, обычно, действующий в фантастической, сюрреальной обстановке или в особой «внеземной» среде) фактически воспринимается играющим как условное неуничтожимое существо. Это также снижает уровень эмоционального сопереживания.

Кроме того, редукция движения персонажа обедняет его характер, зачастую превращая в бездушную механическую марионетку, всегда одинаково падающую, или подпрыгивающую, или становящуюся на одно колено. Так герой лишается той персональной ауры, которая возникает при ручной работе по диснеевскому методу вследствие неповторимости (или достаточного разнообразия и сложности) его движений, мимики, жестикуляции и т.д.

Все это почти полностью отсутствует в диснеевской эстетике, вызывающей сильный эмоциональный отклик, увлекающей зрителя интригой до такой степени, что он буквально может забыть о себе, отождествляясь с персонажем.

Однако необходимо сказать несколько слов и о положительных сторонах «эстетики повышенной условности» компьютерных игр.

Прежде всего эта эстетика замечательна тем, что вводит принципиально иные, новые «правила игры» — например, в

виде неких исключительных свойств экранных персонажей. Эти свойства можно соотнести со свойствами *анимы* — души человеческой, или иначе — нашего сознания: а оно воистину мнит себя неуничтожимым, невесомым, безынерционным, внепространственным, вневременным и т. д. Тем самым в какой-то степени формируется совершенно новая установка психического развития хакеров: они в буквальном смысле становятся «не от мира сего», постоянно общаясь со столь странными «потусторонними» персонажами. Видимо, пока преждевременно рассуждать об эффекте расширения сознания, но можно с большой степенью уверенности говорить о его постепенной, едва начавшейся переориентации, идущей в направлении от земного, физического плана бытия к более тонкому и универсальному по своим свойствам плану — тому, который в оккультизме именуется астральным. Вряд ли следует сразу отмахиваться от этого слегка неожиданного «оккультного следствия» эстетики компьютерных игр — ведь представление о более тонких, неявных планах нашей многомерной вселенной присутствует практически во всех религиозных мировых доктринах (от даосизма и буддизма до христианства и ислама) и имеют весьма почтенную, тысячелетнюю традицию, аккумулируя сокровенный опыт человечества. Вполне возможно, что колоссальный успех и привлекательность компьютерных игр как раз и основаны на «узнавании» (на подсознательном уровне) тех «божественных» состояний, которые каждый из нас переживал в разных перерождениях в иных слоях нашей вселенной. С этой точки зрения, внедрение эстетики Диснея, с достаточной точностью имитирующей законы именно физического плана бытия, способно сильно потеснить ныне

господствующую в области компьютерных игр «оккультную», условно говоря, эстетику. Может ли это привести к преобладанию натуралистических тенденций и к «сдергиванию» хакеров с их компьютерных небес на землю? Оставим этот вопрос открытым.

Совершенно очевидно, что компьютерные игры будут дальше развиваться также и по пути усложнения образов анимации и приближения ее к модели диснеевской эстетики. Однако вряд ли стоит здесь ожидать воспроизведения этой эстетики в полном объеме. Но по крайней мере ее основополагающие принципы, обеспечивающие зрителю максимально благоприятный и комфортный режим восприятия экранных образов, рано или поздно окажутся в поле зрения создателей компьютерных игр. В связи с этим большой интерес для программистов, создающих компьютерные игры, могут представлять излагаемые ниже основные законы диснеевской анимации, разумеется, не исчерпывающие всех принципов этой классической (или иначе — базовой) эстетики анимационного фильма.

12 ПРИНЦИПОВ АНИМАЦИИ ПО ДИСНЕЮ

1. Сжатие и растяжение.
2. Подготовка, или упреждение (отказное движение).
3. Сценичность (постоянный учет того, как видит образ зритель).
4. Использование композировок и фазованного движения.
5. Сквозное движение (или доводка) и захлест действия.
6. Смягчение начала и завершения движения.
7. Дуги.
8. Дополнительное действие (выразительная деталь).
9. Расчет времени.

10. Преувеличение, утрирование.

11. «Крепкий» (профессиональный) рисунок.

12. Привлекательность.

Принципы диснеевской анимации были получены практическим путем, исходя из опыта конкретной повседневной работы. Они оказались столь эффективными, что их изучение стало обязательным сначала для аниматоров Диснея, а потом и для аниматоров всего мира. В целом это наиболее полный набор технологических приемов, необходимых для создания образа и его динамики в анимационном фильме.

Принципы диснеевской анимации имеют гораздо более широкое применение, чем область анимационного кино. Основанные на общих законах восприятия визуальной информации, они обеспечивают наилучший режим восприятия. Фильмы, построенные в соответствии с этими принципами, удобно и комфортно смотреть. Их успех как бы предопределен.

1. **Сжатие и растяжение (Squash and Stretch)** — сквош и стретч — одно из самых важных открытий Диснея. Тем не менее оно практически не используется художниками-аниматорами, создающими современные компьютерные игры. Суть его состоит в том, что живое тело во время движения при каждом шаге (как и при любом действии) то сжимается, то растягивается, то расширяется, то удлиняется вновь. Абрис такого персонажа постоянно «дышит», и это дыхание создает ощущение одушевленности как взаимосвязанности всех частей персонажа. Ощутимо сжимается присевшая фигура — в противоположность вытянутой фигуре в прыжке или в броске. Точно так же утолщается благодаря напряженному бицепсу согнутая рука, становящаяся тоньше (и на вид — длиннее)

при распрямлении. Даже лицо, когда персонаж разговаривает, улыбается или просто меняет выражение, воспринимается живым, только если одновременно, скажем, с движением губ меняется также и форма щек, глаз, подбородка и даже ушей.

Аниматоры Диснея использовали для сквоша и стретча две оси — горизонталь и вертикаль. Наиболее важное правило при этом — неизменность общего «объема» персонажа. Вертикальная растяжка компенсируется горизонтальным сплющиванием, и наоборот. Персонаж уподобляется мешку с мукой: как его ни бросай, количество муки в нем остается постоянным. Такому мешку уподобляются все без исключения диснеевские персонажи. Их тела постоянно «дышат», пульсируют, по их абрисам постоянно проходят плавные волны.

Без сквоша и стретча тело персонажа как бы каменеет. Эта неестественная застылость особенно ощутима в эпизодах схваток противников между собой (в играх «Gang Wars», «Fatal Fury», «King of Fighters», «Ninja Combat» и др.), когда тело побежденного персонажа, сохраняя неизменный абрис, описывает в воздухе дугу и падает на землю, не испытывая при этом никаких деформаций.

2. Подготовка, или упреждение (отказное движение). Перед тем, как сделать любое резкое движение или физическое действие, человеку обычно необходима предварительная подготовка — как бы упреждение действия. Например, перед прыжком вверх приседают, перед прыжком в длину отходят назад; для броска руку с камнем отводят в направлении, противоположном будущему полету камня; перед ударом по мячу ногу отводят назад. Тем самым создается мощное инерционное движение, которое обеспечивает силу для броска, толчка, удара.

Такое подготовительное движение всегда совершается в направлении, противоположном задуманному, поэтому его называют отказным движением: персонаж вроде бы сначала отказывается от своего намерения, чтобы тем вернее его осуществить. В компьютерных играх в основном отсутствует имитация отказного движения персонажей: кулак в драке выбрасывается вперед мгновенно, также мгновенно осуществляется нижняя подсечка противника ногой («King of Fighters», «Street Smart» и др.). Такие безынерционные движения характерны лишь для школ восточных единоборств и частично обретают естественность, если игра имеет соответствующий восточный колорит («Sai Combat», «Ninja Combat» и др.).

Принцип отказного движения действует также и в области человеческой психики. Очень часто для того, чтобы принять, например, новую идею, ее для начала необходимо хорошенько обругать, выявить ее недостатки. Принятию предшествует отрицание, положительным эмоциям — отрицательные, душевному подъему — спад и т. д.

Этот общий принцип отказного движения распространяется в диснеевской анимации на все без исключения действия персонажа и называется также упреждением.

Зритель должен быть подготовлен к каждому последующему движению персонажа и ожидать его еще до того, как оно произойдет. Это достигается упреждением каждого основного движения особым жестом или движением, которое настраивает зрителя на то, что должно произойти. Отсутствие упреждения в компьютерных играх приводит к тому, что действия персонажа распадаются на ряд отдельных движений, их не соединяет ощущение единого инерционного хода, создающего непрерывность восприятия. А возникающее ощущение

дискретности действия персонажа резко обостряет его условность.

3. Сценичность (Staging) — наиболее общий принцип, восходящий к далеким временам в истории театра. Сценичное действие всегда рассчитано на внешнего наблюдателя, на следящего за происходящим зрителя. Поэтому оно стремится к тому, чтобы все было предельно ясным, понятным, узнаваемым. Выражение лица сценично, если оно хорошо читаемо, настроение персонажа сценично, если оно воздействует на зрителя. Характер персонажа должен быть узнаваемым, детали — хорошо заметными, реплики — разборчивыми, текст — доходчивым и т. д.

Движение персонажа не должно скрадываться одеждой, или смазываться неверным выбором угла зрения, или отесняться на второй план чем-то другим. Сценичность особенно важна в компьютерной игре, поскольку сама ее суть предполагает непрерывное слежение игрока за тем, что происходит на экране. Экран здесь — сцена действия. Она должна быть легко обозрима и понятна в своей архитектуре, а само течение игры не должно нарушать эту обозримость. В игре «Marauder» уничтожение боевым самолетом различных целей фиксируется появлением на их месте клуба дыма. При этом на экране постоянно виден сам самолет, его текущее местоположение, что позволяет игроку маневрировать и избегать встреч с пораженными объектами. Однако в самые «горячие» моменты игры клубы дыма на месте пораженных объектов возникают в непосредственной близости от самолета и закрывают его на одну-две секунды, лишая игрока возможности продолжать игру. Часто эти секунды оказываются решающими.

В игре «Cyber Tank» на экра-

не воспроизводится панорамный ландшафт с дорогой, по которой движется танк. Сама дорога и ее обочины насыщены различными объектами: указателями с надписью «Danger», бетонированными дотами, минными полями, проволочными заграждениями; сбоку к дороге выдвигаются танки, гранатометчики, налетают вертолеты и т. д. Несмотря на архитектурную ясность пейзажа, совершенно невозможно контролировать все появляющиеся объекты — скорость их появления, разнообразие и разброс в поле экрана превосходят возможности наблюдателя-игрока. Поэтому двое играющих (при парной игре), воспринимая поле игры в целом и не фокусируя внимание на отдельных объектах, практически непрерывно поливают огнем все поле перед движущимся танком, что является здесь едва ли не единственно возможной, оптимальной стратегией. При этом особо привлекательной оказывается предельная очевидность схемы действия — и одновременно чрезвычайная, «иррациональная» насыщенность игрового поля объектами, трудными для отслеживания еще и за счет имитации изменения угловых расстояний между различными объектами при рыскании танка во время движения.

Иногда сценически выстроить движение можно только с помощью силуэта. Например, если грудь и рука персонажа одного цвета, то рука на груди просто теряется, лишаясь вы-

разительности. Точно так же светлые плечи, поднятые на фоне светлой части головы, сводили на нет жест пожимания плечами. Дисней в этом случае говорил своим аниматорам: «Работайте в силуэте, чтобы все было ясно видно. Не допускайте, чтобы рука заходила на лицо и не было видно, что происходит».

4. Компоновки (Pose to Pose — от позиции к позиции) и фазованное движение (Straight Ahead Action — прямо вперед). Один из аниматоров Диснея утверждает, что работа компоновками была изобретена на студии Диснея. До этого аниматор просто фазовал «прямо вперед», начиная с первого движения персонажа в сцене, последовательно делая рисунок за рисунком, что-то придумывая по мере продвижения, пока не закончит сцену. Аниматору в этом случае известна сюжетная линия, но он не слишком ясно представляет себе, как все это будет выглядеть конкретно. И рисунки, и само действие приобретают при этом способе анимации, импровизационный характер, поскольку в течение всей работы сохраняется элемент неожиданности, неизвестности и новизны.

При работе с компоновками аниматор заранее продумывает, планирует, размечает действие и делает ключевые, наиболее выразительные и сложные фазы движения — компоновки. После этого сцена передается ассистенту для прорисовки промежуточных, более простых

в исполнении фаз. Впоследствии такая оценка воспринимается легко, она «срабатывает», поскольку все взаимосвязи и соотношения просчитаны прежде, чем аниматор с головой уйдет в работу. Основное внимание уделяется отработке самих компоновок и расчету времени движения. При работе с компоновками достигается ясность действия и его напряженность, а при фазованном движении — большая спонтанность, эскизность, легкость. Часто последний подход более эффективен, чем слишком тщательное планирование, поскольку вносит элемент неожиданности.

Кажется, что художник-разработчик компьютерных игр «обречен» на работу с компоновками, поскольку это едва ли не единственное, что остается на его долю, — ведь темп игры, а часто и характер движений, их последовательность задает сам играющий (в персонажной игре). Однако рано или поздно персонажи видеоигр начнут обретать индивидуальные характеры: свою особую и ни на что не похожую походку (как в игре «MoonWalker» Майкла Джексона), характерные жесты, движения и даже привычки (вспомним типажей игрового кино — хотя бы в «Великолепной семерке» или в «Семи самураях»). И тогда вместо набора тщательно рассчитанных компоновок как раз и потребуются та легкость, эскизность и импровизационность образа, которые возникают при фазовке «прямо вперед».

Пример из демонстрационной версии пакета AUTODESK ANIMATOR.



5. Сквозное движение, или доводка (Follow Through), и захлест действия (Overlapping Action) используются для придания движению большей естественности, пластичности и непрерывности. Движение никогда не должно полностью прекращаться до того, как начнется следующее движение. Сквозное движение и захлест обеспечивают непрерывность хода повествования и согласованность отдельных эпизодов.

Доводка предполагает постепенное, неодновременное начало или прекращение движения различных частей тела персонажа в зависимости от их инерционности.

Захлест означает взаимоналожение движений или действий, их пересечение, при котором одно действие постепенно переходит в следующее.

Аниматоры Диснея различали пять основных градаций доводки и захлеста:

- хвосты, уши, длинные плащи и т. п. продолжают двигаться после того, как остальные части тела уже перестали двигаться;

- само тело не двигается сразу, целиком; оно растягивается, сжимается, перекручивается, поворачивается, поскольку все его части работают несинхронно. Если при ходьбе движение начинается с бедер, то после того, как они начинают двигаться, приходят в движение ноги. Бедра являются ведущими, ноги — ведомыми; за бедрами следует торс, затем плечи, затем рука, запястье и, наконец, пальцы. И хотя большинство крупных движений

тела начинается с бедер, движение пальца — с запястья, при движении головы ведущими обычно являются глаза.

Например, кисть или рука могут продолжать двигаться, когда само тело уже остановилось. После остановки руки по ней еще может скользить, скажем, браслет. Для подчеркивания остановки персонажа необходимо, чтобы его голова, грудь и плечи останавливались одновременно, поскольку именно по этим частям тела зритель прежде всего судит о движении персонажа. И лишь затем останавливаются руки и ноги;

- мягкие части тела персонажа (такие, как щеки или тело Дональда, или почти все тело Гуфи) будут двигаться медленнее, чем части, имеющие, условно говоря, скелет. Это отставание в движении называют drag — «оттяжка». Она придает мягкость, неестественность и свободу фигуре, создает ощущение жизненности. Когда это хорошо сделано, то drag практически не заметен при просмотре фильма, воздействуя исключительно на подсознание. Фазы, которые рисует аниматор, и не предназначены для того, чтобы их рассматривали по отдельности, они создают особые динамические эффекты при проекции;

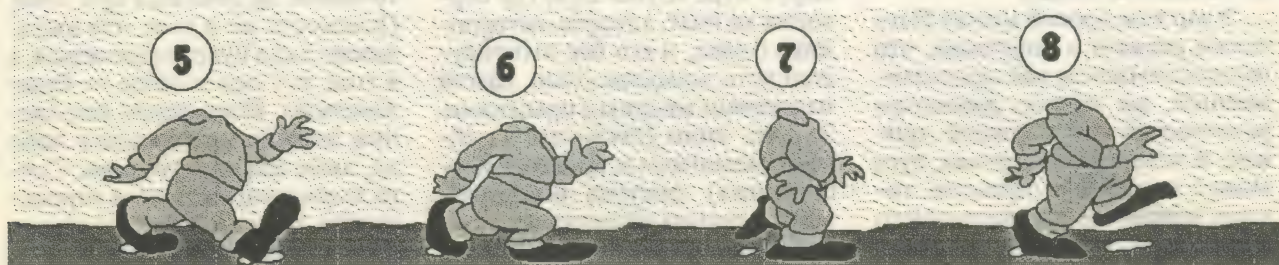
- доводка, или завершение действия — т. е. тщательная отработка концовки любого действия;

- экспрессивная статика (буквально — «движущаяся статика») возникает, если компоновку персонажа без изменения держать на экране на про-

тяжении 8—16 кадров, т. е. менее секунды. Но и этого времени достаточно для восприятия зрителем этой статики как реакции персонажа. Если компоновка удерживалась без движения чуть дольше, терялась иллюзия объема и картинка обнаруживала свою плоскостность. Чтобы этого не происходило, использовалась не одна, а две близкие фазы. Вторая фаза была крайней степенью первой (глаза раскрываются еще шире, уши еще больше выдвигаются вперед, щеки округляются еще больше и т. д.), хотя персонаж остается в том же самом положении). Этот прием усиливает выразительность позы и прибавлял ей живости и экспрессии.

Все эти типы доводки и захлеста практически не используются в компьютерных играх — возможно, вследствие достаточной виртуозности, требуемой в этом случае от художника, который становится здесь также и драматургом, и режиссером.

6. Смягчение начала и завершения движения (Slow In и Slow Out — «медленный вход» и «медленный выход»). При разработке компоновок аниматор добивается их максимальной выразительности и, естественно, хочет сконцентрировать на них внимание зрителя. Для этого таким образом рассчитывается движение от одной компоновки к другой, чтобы основная часть времени пришлась на демонстрацию компоновок. При этом персонаж «промелькивает» от одной компоновки к другой, задерживаясь в них. Такой характер движения и пол-



учил название «медленный вход» и «медленный выход», поскольку при этом промежуточные фазы группируются вокруг компоновок, как бы смягчая, замедляя скорость движений персонажа. В компьютерных играх очень часто в эпизодах поединков движение персонажа представляет собой смену компоновок без каких-либо промежуточных фаз («King of Fighters», «Street Smart», «Final Fight» и др.). Здесь отсутствует не только смягчение, но и вообще начало и завершение движения (выпада, удара кулаком, подсечки ногой и т.д.); они воспринимаются как безынерционные. Точно так же неестественны обычно обладающие сверхподвижностью и мгновенно реагирующие на управляющие воздействия игрока летательные аппараты, уничтожающие те или иные цели («Marauder»).

7. Дуги (Arcs) — использование криволинейных траекторий естественного движения. Живые организмы крайне редко двигаются вверх-вниз или вперед-назад с механической точностью. Голова, например, редко вытягивается сразу вперед, а затем сразу назад; она еще и слегка приподнимается или опускается. Движение дятла — одно из редких исключений.

Открытие движения по дугам произвело настоящий переворот в анимации. Раньше при ходьбе персонажи подсказывали вверх-вниз, как механические игрушки, теперь они описывают дугу в тот момент, когда нога при шагании находится в наивысшей точке и когда находится внизу.

Удар или бросок может быть таким резким и коротким, что воспринимается, как прямолинейный, но начало движения развивается по плавной кривой. В некоторых случаях дуга может перейти в прямую линию, например для падающего предмета. Но обычно, даже ког-

да перемещение идет вдоль прямой, объект подвергается повороту. Сквозное движение, доводка и захлест часто приобретают винтовой характер.

Когда этот принцип движения был понят, все фазы стали размещаться по дугообразным траекториям. Проблема заключается в том, что сделать рисунок по дуге намного труднее, чем просто на полпути между двумя соседними фазами. Даже если на крайних компоновках написано строгое предупреждение «Соблюдай дугу!», все равно остается сильное искушение сделать промежуточную фазу на прямой линии.

При медленном движении, с большим количеством промежуточных фаз, дуга траектории движения достаточно закруглена и выпукла. Если же движение быстрое, дуга спрямляется. Чем быстрее движение, тем прямее дуга. Иногда это предпочтительно, но чаще траектория даже быстрого движения должна быть в виде кривой или дуги. Рисунки, сделанные в промежуточных точках, начисто лишают движения естественности.

Наиболее распространенная ошибка в компьютерных играх — движение насекомых и птиц в полете по прямой, а не по плавной дуге, соответствующей махам крыльев (например, игра «Juju»). Исключением здесь является «хаотический», неровный полет мотылька, образующий петляющую ломаную линию. Точно так же по дуге, слегка поднимаясь и опускаясь, перемещается центр тяжести человека, т.е. его живот, при ходьбе и во время бега. Такая естественная походка удачно воспроизводится, например, в игре «Cyber Lip», где персонаж не идет, а скорее, неторопливо бежит, и его бег имитируется чуть замедленными, слегка парящими мелкими прыжками. За счет этого (несмотря на общий схематизм и простоту изображения) фигурка воспринимается весьма естественно. Если персонаж полностью неподви-

жен во время остановки своего движения (как мы это видим в большинстве игр), он «умирает» для наблюдателя. Удачный выход нашли авторы некоторых игр, где во время остановки персонажа художник имитирует глубокие вдохи-выдохи (грудь поднимается и опускается — игра «Sengoku») или небольшие упругие приседания на полусогнутых ногах («Ninja Combat»). Следует учитывать, что в большинстве восточных единоборств боец, заняв боевую стойку на полусогнутых ногах, при горизонтальных перемещениях удерживает центр тяжести тела на одном и том же уровне — на него не распространяется правило синусоиды («Sai Combat»).

8. Дополнительное действие, или выразительная деталь (Secondary Action). Часто подкрепляет идею, воплощенную в сцене: печальный персонаж долго сморкается, отвернувшись; смущенный человек протирает очки, достает из портсигара папироску, поправляет галстук. Подобные детали называются вторичным, или дополнительным действием, которое всегда подчиняется главному. Если вторичное действие вступает в конфликт с основным или становится более интересным, значит неверно выбрана выразительная деталь или постановка не сценична.

Иногда вторичным действием является выражение лица. Когда основная идея передается движением тела, выражение лица становится подчиненным этой идее. Например, персонаж переходит от смеха к сосредоточенности, двигаясь при этом. Опасность не в том, что выражение лица будет выделяться, а в том, что оно вообще не будет замечено. Изменение выражения должно происходить либо до, либо после движения. Смех на выражения во время перемещения вряд ли будет замечен, и его смысл потеряется.

Отсутствие дополнительных действий приводит к схематизации персонажей, их превращению в функциональных роботов, по отношению к которым можно испытывать определенный азарт, но не сопереживание. Случайные, незапланированные мелкие движения персонажа или изменения выражения его лица, вздохи, короткие реплики сразу придают индивидуальность герою и выделяют его среди прочих персонажей.

9. Расчет времени (Timing)

— определение интервала между действиями для подчеркивания веса, размера и характера персонажа. Число фаз, использованных для любого движения, определяет то время, которое данное действие займет на экране. Расчет интервалов времени, с помощью которого выбирается скорость движения, очень важен, так как он придает смысл движению. Выбором скорости действия определяется, насколько идея, заложенная в действии, будет понята зрителем. При расчете времени учитывается вес и размер объекта, а также его внутреннее состояние.

Верный расчет времени — ключевой момент для восприятия экранных образов зрителем. Важно задать достаточно времени, чтобы подготовить зрителя к ожиданию действия, самому действию и реакции на действие. Если выделяется слишком много времени, то внимание зрителя рассеивается. Если времени мало, действие может закончиться, прежде чем зритель его заметит, и смысл действия будет утрачен.

При ускорении движения очень важно быть уверенным в том, что зритель сможет уследить за происходящим. Действие не должно быть настолько быстрым, чтобы остаться за порогом осознания.

Более, чем другие принципы, расчет времени задает вес и размер объекта. Два объекта,

одинаковые по размеру и форме, могут казаться разными по весу, если менять хронометраж их движения.

Требуется много сил, чтобы заставить перемещаться пушечное ядро, но, начав движение, оно продолжает двигаться с постоянной скоростью, и нужно приложить много усилий и времени, чтобы остановить его. Легкие объекты требуют меньше времени для преодоления своей инерционности. Достаточно задеть пальцем воздушный шарик, чтобы он отлетел в сторону. Имея малую инерцию, он быстро останавливается даже из-за трения о воздух. Расчет времени влияет также на передачу размера и масштаба объекта или персонажа. Великан имеет больший вес, большую инерцию, чем обычный человек. Поэтому он сдвигается с места медленнее. Любое изменение в движении для него происходит медленнее. Наоборот, маленький персонаж имеет меньшую инерцию, чем обычный, и его движения должны быть более быстрыми.

Манера поведения объекта на экране, ощущение его веса целиком зависят не от прорисовки отдельных фаз, а от тайминга промежутков между ними. Не слишком важно, как изображено пушечное ядро. Оно все равно не будет похоже на пушечное ядро, если не будет вести себя в соответствии с присутствующим ему весом и инерционностью. То же самое верно для любого объекта и персонажа.

Эмоциональное состояние персонажа также больше определяется его движениями, чем внешностью. А изменение скорости движения может показывать, что персонаж утомлен, возбужден, нервничает или отдыхает. Старейшие аниматоры Уолта Диснея Томас Фрэнк и Олли Джонстон приводят пример того, как изменение продолжительности одного и того же действия меняет его значение. Всего лишь два рисунка го-

ловы, первый — изображающий наклон головы к левому плечу, а второй — к правому, могут передавать огромное количество разных состояний, полностью зависящих от того, какая продолжительность движения была использована. Каждая промежуточная фаза, добавленная между крайними положениями, придает действию новый смысл:

- промежуточных фаз нет — персонаж получил удар страшной силы; ему почти снесло голову;

- одна промежуточная фаза — ударили кирпичом, скалкой, сковородкой;

- две промежуточных фазы — у персонажа нервный тик, спазмы мышц или судороги;

- три промежуточных фазы — персонаж уклоняется от удара кирпичом, скалкой, сковородкой;

- четыре промежуточных фазы — персонаж кивком головы отдает отрывистый приказ типа «Иди!», «Толкай это!»;

- пять промежуточных фаз — персонаж более дружелюбен: «Сюда, пожалуйста» или «Давайте поскорее»;

- шесть промежуточных фаз — персонаж увидел привлекательную девушку или спортивный автомобиль, который ему всегда хотелось иметь;

- семь промежуточных фаз — персонаж старается что-то лучше разглядеть;

- восемь промежуточных фаз — персонаж ищет ореховое масло на кухонной полке;

- девять промежуточных фаз — персонаж оценивающе задумался;

- десять промежуточных фаз — персонаж потягивает ноющие мышцы.

В компьютерных играх достаточно часты случаи неудачного или неумелого тайминга. Например, в «Cyber Tank» при подрыве танка его башня, сорванная взрывом, при ударе о землю подпрыгивает, словно пластиковая, вызывая сомнения в ее массивности. В различ-

ных играх со схватками посылаемые в нокаут персонажи, грохаясь на землю, то сразу как бы прилипают к земле («Double Dragon 3. The Rosetta Stone»), то наоборот, подпрыгивают, словно гуттаперчивые («Final Fight»), иногда при этом вздымая, как в комиксе, аккуратные круглые клубы пыли, фиксирующие падение («Burning Fight»). Порой неправильный тайминг приводит к тому, что у шагающего персонажа наблюдается проскальзывание ног при ходьбе («Street Smart»). В анимации это происходит при несоответствии скорости протяжки панорамы и скорости движения персонажа. Такая ошибка особенно бросается в глаза, когда останевившийся персонаж по-прежнему продолжает шагать («Sengoku», «Burning Fight»). В игре «MoonWalker» Майкла Джексона можно наблюдать парадоксальную ситуацию, когда смещающийся назад персонаж продолжает шагать вперед — тайминг его шага остается неизменным для сохранения общего пластического рисунка роли и не учитывает смены направления движения. Иногда встречается другая крайность: останевившийся персонаж скользит, смещаясь параллельно самому себе, не переступая ногами — т.е. не шагая («Ninja Combat»). Досадны моменты неточного сведения персонажа и фона. Например, в «Sengoku» один из поединков происходит на спинах лошадей, бегущих сплошной лавиной. И все же в какой-то момент нога одного из персонажей оказывается висющей в пустоте, выйдя за пределы общего абриса табуна. При этом особенно нелепой выглядит черная полоска тени от тела, по-прежнему соединяющая обе ступни персонажа, — т.е. выглядит так, словно она падает на ровную горизонталь. В игре «Shadow Dancer», где действует ниндзя с собакой, когда персонажи поднимаются вверх, прыгая со ступени на ступень по наваленным ящикам, в один из мо-

ментов игры собака не уместается на узкой ступени и ее задние ноги, висая в пустоте, как бы опираются на невидимую опору. Такие моменты сразу уничтожают иллюзию правдоподобия и ставят играющего перед фактом обостренной условности происходящего.

10. Преувеличение, утрирование — выделение главного в замысле через постановку и действие. Диснеевские аниматоры вспоминают, что Уолт требовал от них больше реализма, при этом на самом деле стремясь к «карикатурному реализму». Уолт говорил, что если персонаж должен быть печальным, сделайте его мрачным; счастливый — пусть будет ослепительно сияющим, беспокойный — раздраженным, неистовый — необузданным. Под утрированием Дисней понимал выделение и подчеркивание главного в замысле через постановку и действие. В анимации утрированное, преувеличенное больше говорило зрителю — аниматор подчеркивает главные черты, выделяет суть любого объекта так, чтобы зрители это восприняли.

Для образов анимации характерны воздействия в диапазоне слабых эмоций (интереса, лирики, стыда). С помощью преувеличения в определенной степени достигалось усиление эмоционального воздействия. Этим же путем идут сегодня использующие диснеевскую эстетику режиссеры компьютерной анимации. Например, Джон Лассетер, описывая работу над фильмом «Люксо-младший», где две настольные электрические лампы играют в мячик, пишет, что движения персонажей-ламп в фильме передают ощущение реальной предметности, хотя почти каждое перемещение утрировано, чтобы подчеркнуть его. Скажем, когда лампа подскакивает для прыжка, ее движения преувеличены для передачи ощущения веса

подставки. На звуковой дорожке записаны скрипы реальной лампы. Затем, чтобы подчеркнуть отдельные действия, добавлены утрированные звуки. Цель иронического эффекта всех этих преувеличений, как пишет Джон Лассетер, — сделать фильм более естественным и в то же время занимательным. Как можно видеть, компьютерная анимация легко освоила диснеевские принципы и успешно применяет их, реализуя на совершенно иной технологической базе, нежели целлулоидная мультипликация.

11. «Крепкий» (профессиональный) рисунок. Один из аниматоров Диснея, подчеркивая важность высокопрофессионального владения искусством рисования, говорил: «Рисовать — это все равно что устраивать представление, художник — это актер, который не стеснен рамками собственного тела и которого ограничивает только умение и, может быть, опыт». Одна из табличек, висевших на студии Диснея, вопрошала: «Чувствуется ли в твоём рисунке вес, глубина и равновесие?», напоминая об основах классического рисунка.

Другая табличка предостерегала от появления «близнецов» в рисунках — когда обе руки или обе ноги персонажа не только симметричны, но и делают одно и то же. Никто не делал этого нарочно, обычно художник даже не предполагал, что у него так получится.

В некоторых компьютерных играх близнецы вводятся намеренно — в виде совершенно одинаковых персонажей, которые одинаково ходят, машут кулаками, падают и т. д. Иногда персонажи-близнецы различаются только цветом: например, синий и красный ниндзя («Ninja Combat») в парной игре.

Для аниматора зачастую недостаточен опыт художника комиксов. Ведь рисунок в анимации важен не сам по себе, но как

элемент движущегося изображения. Здесь от рисунка требовалась форма, которая являлась бы «живым телом», способным двигаться — в противоположность статичным изображениям. Для такого рисунка подходило определение «пластичный». Оно как бы заключало в себе ощущение внутренней текучести и динамики.

12. Привлекательность (Appeal). Привлекательным может быть любой предмет, если на него смотришь с удовольствием, обнаруживая в нем обаяние, простоту, хороший дизайн, понятность и притягательность, очарование и магнетизм.

Персонаж, обладающий привлекательностью, привлекает и удерживает взгляд. Дисней считал, что привлекательным должен быть любой персонаж: злодей, пусть даже страшный и трагический, все равно должен обладать привлекательностью, иначе вам не захочется смотреть, что он делает. Уродливый, отвратительный персонаж должен притягивать взгляд вне зависимости и связи с ситуацией и со своим характером.

Не хватает привлекательности обычно слабому или сложному и чересчур запутанному рисунку. Привлекательными могут быть выражение лица, персонаж, движение или вся ситуация в целом, весь сюжет. Однако если, например, аниматор будет стараться передать слишком тонкие оттенки в выражении лица персонажа, рисунок может стать плохо воспринимаемым.

Следует сказать также об огромном влиянии экономической стороны производства на эстетику диснеевского фильма. Все те приемы, которые значительно увеличивали стоимость производства фильма, автоматически становились неупотребительными. Аниматоры Диснея писали: «Можно добиться многих великолепных эффектов, но слишком часто оказы-

вается, что они стоят дороже, чем в среднем можно себе позволить при производстве <...> В середине 30-х годов мы мечтали о фактурном рисунке без обязательного линейного контура, но тогда это было непрактичным. Нам приходилось искать другие пути для воплощения нужных моментов в сцене». Заметим, что бесконтурный рисунок и сейчас по-прежнему остается «непрактичным», и его могут позволить себе лишь создатели «авторских» фильмов, которых не заботит коммерческий успех их продукции и проблемы рынка.

В заключение приведем любопытный пример выбора «астрального» игрового сюжета («Sengoku»), который, во-первых, не требует воспроизведения основополагающих феноменов земного существования, во-вторых, делает просто необходимым все те «паранормальные» феномены, которые в обычной игре оказываются не слишком желательными моментами условности. В «Sengoku» воспроизводится богатый спектр магических действий: материализация и дематериализация персонажей («убитый» персонаж уменьшается в размерах, стягиваясь в точку и исчезая, а затем вновь в потоке энергии «воплощается» в полупрозрачном «астральном» теле, постепенно приобретающем плотность); трансформации в иной облик; перемещение из одного игрового поля в другое (из прошлого в настоящее и обратно) в потоке энергии; использование в качестве оружия энергетических эманаций различного вида, излучаемых персонажем (энергетические вихри, смерчи, полумесяцы). Вместо условной «смерти» астральные противники вновь превращаются в энергетический сгусток (голубое пламя, вспышка), втягивающийся под землю; с помощью особых «пузырей» визуализируется пред-

ложение героя магической помощи, которую он по своему усмотрению может принять или нет. Здесь вполне естественными выглядят такие противники, как скелеты или гигантский человеческий череп. Эта игра концентрирует в себе те магические возможности, которые являются специфическими для эстетики компьютерного искусства в целом и в случае «астральных» сюжетов мощно проявляют свой потенциал, вырвавшись за узкие пределы имитационных и в конечном итоге натуралистически жизнеподобных игр. Когда-то Циолковский предрекал человеку существование в виде сгустка энергии, свободно распространяющегося в мировом пространстве. Компьютерное искусство нашего времени, зачастую само не подозревая об этом, тяготеет к изображению этого далекого и столь трудно представимого будущего человечества, овладевшего приемами «магии», текуче изменяющего свой физический облик и в этих трансформациях выявляющего свою неизменную, «тонкую» энергетическую природу.

Л и т е р а т у р а

1. Джон Лассетер. Принципы традиционной мультипликации и их применение в трехмерной компьютерной мультипликации // Графика-кон'91. Компьютерная графика в науке и искусстве. Материалы конференции. — М.: АН СССР и ACM SIGGRAPH, 1991, с.50—65.
2. Frank, Tomas and Johnston, Ollie. Disney Animation. The Illusion of Life. New York, Abbeville Press, 1981, p. 47—69.

ОБ АВТОРЕ

Алексей Михайлович Орлов — канд. искусствоведения; преп. Российского кинолицея; научн. сотрудник Аниматографического центра ПИЛОТ; автор более 40 публикаций по эстетике анимации. Контактный телефон: (095)169-85-71.

Своя игра

Андрей Родионов

«ИГРА — вид непродуктивной деятельности, мотив которой заключается не в ее результатах, а в самом процессе. В истории человеческого общества переплеталась с магией, культовым поведением и др. ... Свойственна также высшим животным».

Советский энциклопедический словарь.

С тех пор, как я начал заниматься программированием, мне не раз приходилось сталкиваться с различными компьютерными играми — такими, которые могли общаться с играющим только при помощи неоновых лампочек на пультах старых ламповых ЭВМ, и играми на суперсовременных компьютерах, больше похожими на интерактивный художественный фильм с элементами мультипликации, сложными видеоэффектами и многоканальным звуковым сопровождением.

Многое захватывает в таком творчестве. И не сам процесс игры, а разработка игровой вселенной, ее проектирование и реализация. Когда можно слить воедино сценарий, графику, музыку, искусно задуманный и умело запрограммированный алгоритм — создать единый фантастический мир, живущий по законам, которые ты же для него и придумал.

Можно почувствовать себя почти всемогущим. Можно даже вообразить себе, что ты стал немного ближе к нашему Творцу и стал чуть лучше понимать его. «Limitation is only Your Imagination!» («Ограничение — только ваше воображение!») Этот лозунг, использованный в свое время для рекламы музыкальных синтезаторов, на мой взгляд, подходит и для разработки игр (если у разработчика, конечно, достаточно мощный компьютер).

...И НА ДУДЕ ИГРЕЦ

Автор этой статьи, Андрей Родионов, хорошо известен не только как опытный разработчик системного и прикладного программного обеспечения с более чем двадцатилетним стажем работы на компьютерах различных поколений и моделей, но и как талантливый музыкант, композитор, давно и успешно работающий в различных музыкальных жанрах. Андрей первым в нашей стране записал несколько пластинок с использованием экспериментальных компьютерных технологий. Им написано много статей и для музыкантов, и для программистов, одна из которых была опубликована в самом первом номере нашего журнала. Но не только его музыка и статьи изве-

За долгие годы накопилось довольно много мыслей о программировании в целом и о разработке компьютерных игр в частности. Одни из них уже обсуждены в беседах с друзьями и коллегами, другие легли в основу этой статьи. Спасибо всем, кто терпеливо меня выслушивал и иногда давал советы.

КЛАССИФИКАЦИЯ КОМПЬЮТЕРНЫХ ИГР

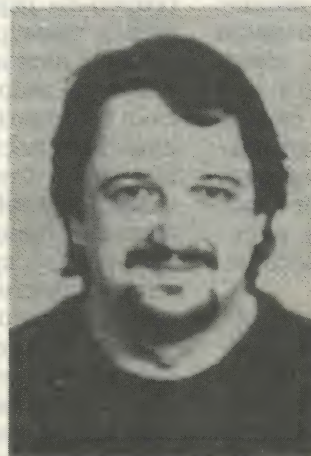
Авторская классификация

Современные компьютерные игры я разделю бы прежде всего на два больших класса:

- для игровых автоматов;
- для индивидуального (домашнего) пользования.

Начнем с игр для игровых автоматов. Игровым автоматом будем называть устройство, предназначенное для сбора денег за предоставляемую возможность играть с этим устройством. Такие автоматы обычно устанавливают в доступных местах или специальных игровых залах. Заметим, что к этому классу не относятся специализированные игровые компьютеры, приобретаемые для индивидуального домашнего использования.

Игры, предназначенные для игровых автоматов, отличаются в первую очередь тем, что для



стны нашим читателям. Многие пользуются его системными и прикладными программами, играют в созданные им компьютерные игры, а одна из первых больших отечественных информационных систем в Академии наук, для которой он проектировал и создавал программное обеспечение с 1975 по 1986 год, успешно эксплуатируется до сих пор. В настоящее время А. Родионов продолжает заниматься как сочинением музыки,

так и практическим программированием на персональных компьютерах различной архитектуры, отдавая предпочтение созданию игровых программ и инструментов для их разработки.

накопления возможно большего количества монет (жетонов) число играющих с автоматом людей (или попыток одного и того же человека играть многократно) должно быть также как можно больше.

Таким образом, игры для этих автоматов должны осуществлять жесткое «давление» на игрока по времени и активную стратегию на «выбрасывание» играющего из игры. При этом сложность каждого нового этапа игры должна возрастать настолько, чтобы среднее время даже для опытных играющих не становилось слишком большим.

Кроме того, игры этого класса должны удовлетворять и некоторым другим требованиям:

- * первая стадия игры должна быть относительно простой и одновременно привлекательной для новичков, но при этом и весьма непродолжительной — новичок должен «выбрасываться» достаточно быстро, а опытный игрок — не задерживаться надолго;

- * логическая схема игры должна быть реализована «жестко» — практически без элементов случайности, так, чтобы играющий во время повторной игры мог использовать свой опыт прохождения всех предыдущих стадий игры для достижения новой;

- * игра должна «поощрять» игрока при завершении очередного этапа — оригинальной графикой, музыкой, а при достижении финала или важных промежуточных стадий — «призовой игрой». Это важно также и для привлечения и поддержки «болельщиков», которые обычно находятся рядом с играющим и являются потенциальными новыми игроками;

- * игра должна быть популярной длительное время, так как стоимость игрового автомата выше стоимости персонального компьютера, а частая смена работающих в нем программ обычно не предусматривается. Некоторые автоматы выпускаются с управлением, рассчитанным только для какой-либо конкретной игры, например, кабина автомобиля или самолета с имитатором наклона кабины, штурвалом, педалями и тремя экранами, расположенными панорамно. Игры в этом случае обычно хранятся не на магнитных носителях, а в ПЗУ.

Чаще всего, перед тем как попасть в игровой автомат, базовая версия программы проходит «обкатку» в качестве игры для домашнего пользования, и только имея устойчивый высокий

рейтинг в компьютерных журналах и пройдя специальную экспертизу фирм-изготовителей игровых автоматов, игра попадает туда, куда (возможно) и предназначалась первоначально. Окончательный вариант игры (т.е. уже для автомата) обычно обладает более изощренной графикой, звуком и другими эффектами, которые отсутствуют в базовом варианте игры для персональных компьютеров.

Однако фирмы-изготовители игровых автоматов разрабатывают и свои собственные специальные игры (иногда вместе с уникальной аппаратурой для них), которые не встречаются на персональных компьютерах и существуют только в версиях для игровых автоматов.

К классу игр для игровых автоматов относятся, в первую очередь, так называемые «аркадианские» игры (Arcadian games), в которых

приходится много стрелять из бластеров, пушек, сражаться различными способами со всевозможными врагами или спастись от разнообразных монстров, пытающихся вас съесть. Действие может разворачиваться в любом времени или месте или вообще неизвестно где и когда. Цель всегда благородна: спасти кого-то (но, в первую очередь, самого себя), победить врага, пытающегося завоевать, разрушить, съесть и пр.

Игры этого класса могут быть выполнены с использованием великолепной графики, обладать прекрасным музыкальным и речевым сопровождением и выглядеть почти как художественный фильм, но при этом — все равно являться играми, где надо все время «спасаться» и «побеждать», причем в том темпе, который диктует компьютер.

Во вторую очередь, я отношу к классу «игр для игровых автоматов» абстрактные игры типа Распан и Tetris, которые, тем не менее, подчиняясь «закону игрового автомата», постепенно уменьшают время на обдумывание ситуации и сводят время реакции играющего к времени опроса компьютером органов управления игрой. Темп игры здесь также диктуется компьютером. Когда время реакции опытного играющего становится соизмеримым с временем опроса органов управления игровой программой, компьютер начинает реализовывать свою собственную выигрышную стратегию в ущерб приоритетности выполнения команд игрока. И тогда у игрока возникает законная реакция — «Так не честно!».

Наконец, среди игр для игровых автоматов

Игровые автоматы должны осуществлять как можно более жесткое «давление» на игрока по времени и активную стратегию на «выбрасывание» играющего из игры, так как их основное предназначение — собрать как можно больше денег.

попадают выполненные как высококачественные *трехмерные тренажеры* (в частности, уже упоминавшиеся автомобили и самолеты), требующие, в отличие от предыдущих игр, высокопроизводительных центрального и видеопроцессоров, специальной аппаратуры и органов управления. Обычно они в меньшей степени подчинены стратегии скорейшего «выбрасывания» игрока, так как требуют более продолжительного освоения. Но это скорее исключение, чем правило. Постоянный спрос на такие игры поддерживается в основном из-за того, что они практически невоспроизводимы на обычных персональных компьютерах.

Таким образом, игры для игровых автоматов (независимо от того, реализована игра действительно на игровом автомате или на универсальном компьютере) характеризуются тем, что доминантой в них является возрастающие темп и сложность игры, которые постоянно диктуются компьютером. При этом вся воля и все умение игрока подчиняются одной-единственной цели — «как можно дольше продержаться».

Время для обдумывания и выбора стратегии и тактики игры практически отсутствует, да оно на самом деле и не требуется, так как каждая ситуация жестко «защита» в самой игре и в точности повторяет сама себя при повторении игры с самого начала. Одно может помочь играющему — опыт преодоления предыдущих стадий при многократных попытках доиграть в такую игру до конца.

Здесь я позволю себе небольшое личное замечание: игры этого класса никогда меня не только не привлекали, а скорее наоборот — отталкивали, так как любое давление будило во мне протест. А в этих играх на меня давил мой собственный компьютер!

Для себя я условно называю эти игры «злыми» по отношению к играющему.

Перейдем теперь ко второй группе: «игры для индивидуального (домашнего) пользования».

Игры этого класса, которые я условно называю «добрыми» по отношению к играющему, гораздо более разнообразны, чем предыдущие. Однако во избежание недоразумений сразу исключим из этого класса все игры, относящиеся к разновидности «злых», хотя вы, может быть, играли в них на персональном компьютере, а вовсе не на игровом автомате. Может быть, эти игры еще проходят «обкатку» или просто несколько хуже других «злых»? А может быть, они уже есть в игровых автоматах, только вы об этом не знаете?

Напомню еще раз, что «злые» игры легко отличить от «добрых» по степени «давления», оказываемого ими на играющего; в «злых» играх «давление» — самоцель и доминанта всей игры,

так как ее истинной задачей является как можно более частая смена игроков для сбора денег; в «добрых» — лишь одно из множества средств для взаимодействия с играющим.

Конечно, и в «доброй» «домашней» игре могут быть ограничения по времени, по каким-либо ресурсам или в виде активно нападающих врагов, но это «давление» почти всегда можно смягчить, а иногда даже и полностью его избежать, так что игра может продолжаться длительное время, давая возможность размышлять, спокойно принимать продуманные решения, а иногда и сохранять на внешних носителях промежуточные стадии игры для того, чтобы позже продолжить ее с желаемого момента.

Общепринятая классификация

Приводимое ниже деление игр на типовые (жанровые) классы, хотя и используется достаточно широко во всем мире, является весьма условным, так как многие игры обладают сразу несколькими классификационными признаками, а иногда включают элементы обучения, программирования, разнообразные средства мультимедиа и многое другое. При рассмотрении типовых (жанровых) классов попытаемся спроектировать их на классы «добрых» и «злых» игр.

В условном классе «добрых» игр находятся практически все так называемые «приключенческие» игры (Adventure), где цель известна, но не известны конкретные способы ее достижения. Обычно играющему предоставляется возможность исследовать окружающий его мир, подбирать и использовать различные предметы, общаться с окружающими его живыми и неживыми существами и т.д. Естественно, что такая исследовательская деятельность не терпит суеты и спешки, требует времени, внимания, размышлений и, следовательно, никак не может сочетаться с функциональным назначением игрового автомата (собрать как можно больше денег).

Разновидностью «приключенческих» являются «ролевые» (от слова роль) игры (RPG — Role Playing Games). В них играющий может выступать в качестве любого персонажа игры, а за остальных в это время играет компьютер. Особо популярны в качестве сюжетов для таких игр различные детективные расследования.

Интересным подклассом в классе «добрых» игр являются различные «стратегические» игры (Strategy Games), из самого названия которых уже следует, что играть в них надо не торопясь и хорошо продумывая свою собственную стратегию. Примером могут служить стратегические планы ведения войн, разработки экономической политики и др.



«Симуляции» (Simulations) включают в себя моделирование как относительно небольших объектов (катеров, подводных лодок, самолетов, космических кораблей), так и более крупных (отдельных предприятий, городов, стран, планет и целых планетных систем). Во всех случаях играющему предоставляется возможность добиваться каких-либо целей путем определенных воздействий на различные системы, обладающие сложным поведением, и наблюдать за их реакцией в ответ на такие воздействия. Эти игры также обычно требуют к себе внимательного и вдумчивого отношения и чаще всего попадают в класс «добрых».

«Спортивные» игры (Sport Games) часто требуют хорошей реакции и быстроты действий играющего, поэтому оказываются в зависимости от реализации и среди «добрых» и среди «злых» игр. Примерами могут служить всевозможные «олимпиады», «многоборья», не слишком сложные «авто- и моторалли», «горные лыжи», «теннис», «гольф» и др.

Компьютерные реализации традиционных настольных игр (Board Games) чаще всего принадлежат к классу «добрых» (шахматы, го и т.п.). Карточные игры в зависимости от степени сложности и конкретной реализации могут попадать и в «добрый», и в «злой» классы.

Головоломки (Puzzles) чаще всего принадлежат к классу «добрых» игр, так как обычно предполагают определенные размышления.

Абстрактные игры типа Tetris с одинаковым успехом могут попадать в оба класса в зависимости от выбранных критериев окончания игры.

Уже упоминавшиеся «аркадианские» игры практически полностью относятся к классу «злых».

Обучающие игры (Learning Games) могут оказаться в любом из приведенных выше типовых классов, но при этом они преследуют конкретную цель — научить играющего чему-либо полезному: быстрому счету, чтению электронных схем или дорожных знаков, иностранному языку, печатанию на машинке, оптимальному взаимодействию с деловыми партнерами или учреждениями и др. Конечно же, почти все обучающие игры принадлежат к классу «добрых», так как предполагают вдумчивое и целенаправленное использование их для обучения. Отдельный подкласс обучающих игр образуют «тренажеры», выполняемые чаще всего в типовом классе «симуляции».

Приведенная выше классификация не претендует на полноту, но позволяет легче ориентироваться в новых, появляющихся на рынке играх. И все-таки следует еще раз обратить внимание читателя на то, что в современных сложных компьютерных играх часто соседствуют и могут пересекаться элементы игр разных классов, что затрудняет их точную классификацию.

ОСОБЕННОСТИ РЫНКА ИГРОВЫХ ПРОГРАММ

Рынка игр для игровых автоматов практически не существует, поскольку игры для них, как уже было замечено выше, обычно появляются, продаются и испытываются в первую очередь на рынке программ для ПК.

Чем же этот рынок интересен для разработчика компьютерных игр? Прежде всего тем, что он практически ненасыщаем. Большинство игр рано или поздно надоедают играющим, и в них перестают играть. На смену им приходят новые игры, которые, в свою очередь, вытесняются следующим поколением игр, и так далее...

Неотвратимость появления новых игр гарантируется, с одной стороны, необходимостью создания все новых и новых развлечений для заполнения досуга людей, а с другой стороны — появлением все более совершенных и мощных персональных компьютеров, позволяющих даже

старые игровые идеи поворачивать новой и подчас неожиданной стороной, обогащать дополнительными эффектами, а иногда и наполнять совсем другим содержанием.

Естественно, что разработчиков компьютерных игр устраивает такое положение дел, поскольку оно всегда гарантирует спрос на их работу при условии конкурентоспособности их продукции и соответствующей охраны авторских прав.

Помимо этого, опытные разработчики оказываются заинтересованными также и в том, чтобы их продукция после завершения основного этапа продаж каждой отдельно взятой игры не блокировала продажу следующих за ними новых игр, то есть **устаревала бы достаточно быстро**. Для этого значительное количество игр сознательно проектируется так, что, будучи единожды пройденными от начала до конца, они в значительной степени теряют свою привлекательность из-за ставшей уже известной оптимальной последовательности действий. Этого добиваются обычно увеличением количества «жестко» запрограммированных ситуаций и исключением или сведением к минимуму элементов вероятностного порождения мира игры.

Покупатель же, со своей стороны, хочет, чтобы купленная им игра (как и любая другая приобретенная вещь) не теряла для него своей привлекательности как можно дольше!

Здесь, как мы видим, интересы разработчика игр и играющего существенно противоречат друг другу.

Это противоречие может смягчаться или отсутствовать в следующих случаях:

- если разработчик данной игры не относится к разработке компьютерных игр как к источнику постоянного заработка или вообще не рассматривает данную игру в качестве коммерческого продукта (разработчик-непрофессионал).

- если разработчик-профессионал делает игровую программу в первую очередь для самого себя.

Первая ситуация (разработчик-непрофессионал) встречается довольно часто. Каждый из нас может ознакомиться с образцами таких любительских игр, просматривая различные public domain (программы для бесплатного публичного копирования и использования). Обычно такие игры обладают маловыразительными графикой

и звуковым сопровождением, недостаточно удобны в управлении, а иногда обладают и непонятным сценарием.

Вторая категория игр (разработчик-профессионал делает игру для себя) встречается сегодня крайне редко. Мне известно всего лишь несколько таких игр. Эти игры отличаются прежде всего тем, что, обладая великолепными художественными и алгоритмическими решениями, они при каждом новом запуске предлагают играющему достаточно богатую, но несколько отличную от предыдущей игровую вселенную.

При этом детали этой вселенной каждый раз немного изменяются случайным образом. Это очень точно взвешенное соотношение детерминизма и случайности иногда может создавать такое сочетание разнообразия и неожиданности, которое способно сделать игру практически «бессмертной».

Однако соблюсти этот тонкий баланс, придающий игре «бессмертные» свойства в глазах не только автора, но других играющих, зачастую бывает нелегко и профессионалу, так как большинство игр со сложной и интересной игровой вселенной весьма склонны к вырождению либо в слишком запутанные и чересчур сложные для освоения другими игроками, либо в «хаотические».

Вообще говоря, разработка игровых вселенных связана, на мой взгляд, с очень серьезными философскими и религиозными проблемами, но об этом я скажу несколько позже. А пока зададим себе вопрос:

С КЕМ ИЛИ С ЧЕМ МЫ ИГРАЕМ?

Ответ может быть тривиальным: «С компьютером!».

Или чуть сложнее: «С программой, реализующей алгоритм игры».

Но я предпочитаю развернутый и, как мне кажется, более содержательный ответ: «Каждый раз мы играем с некоторой вселенной, сконструированной авторами игры, которая состоит из объектов, участвующих в данной игре. Эти объекты подчиняются определенным законам, описывающим их поведение, и могут взаимодействовать как друг с другом, так и с играющим. Законы и объекты этой вселенной описываются

*Каждый раз мы играем
с некоторой вселенной,
сконструированной авторами
игры. Законы и объекты этой
вселенной описываются
алгоритмами и данными,
совокупность которых
реализована в виде игровой
программы.*

алгоритмами и данными, совокупность которых реализована в виде игровой программы».

Несмотря на кажущуюся сложность такого ответа, он применим к любым компьютерным играм, и каждый может убедиться в этом сам, вспомнив известные ему игровые программы.

Но его особая ценность, на мой взгляд, заключается в том, что с его помощью можно попытаться разобраться в том, чем же является компьютерная программа вообще и игровая программа в частности.

ЧТО ЖЕ ТАКОЕ КОМПЬЮТЕРНАЯ ПРОГРАММА?

Для начала попробуем понять, чем принципиально отличается практически любая компьютерная программа, предусматривающая ввод (управление), обработку и вывод (отображение) данных, от других продуктов человеческого творчества.

В первую очередь, необходимо отметить, что творческие процессы в любой области человеческой деятельности имеют много общего. Любая творческая деятельность в области науки, литературы, изобразительного искусства, кино или музыки состоит из некоторых «озарений», всегда появляющихся непредсказуемо и внезапно, и из систематического «конструирования» и синтеза целого из этих отдельных «озарений», которое называется мастерством или школой. В разных произведениях даже одного и того же автора соотношение элементов «озарений» и «конструирования» может быть различным. И все же не оно является критерием ценности конечного продукта творчества — ценность определяется цельностью и внутренней непротиворечивостью нового произведения, будь то литературная поэма или новая физическая теория. В любом случае происходит «отчуждение» результата работы духа и интеллекта автора в некоторый конечный продукт, доступный для понимания и употребления другими членами человеческого сообщества.

Не является в этом смысле исключением и одна из самых молодых областей приложения творческих способностей человека: область разработки программ для компьютеров. Работа в

этой области точно так же состоит из «озарений» и «конструирования», качество конечного продукта зависит от мастерства автора и представляет тем большую ценность для окружающих, чем более цельным и внутренне непротиворечивым будет результат работы. Точно так же в конечном продукте присутствует «отчуждаемая» часть работы духа и интеллекта автора, но имеется и одно, очень существенное отличие: «отчужденный» продукт, применительно к компьютерным программам, обладает поведением, которое автор описывает в виде алгоритма работы программы, предусматривая ее реакцию на разнообразные внешние воздействия, в том числе память о результатах предыдущих воздействий по времени (конечный автомат).

Это единственное отличие компьютерных программ от других продуктов творческой деятельности человека является, тем не менее, принципиальным, так как до этого человеку крайне редко предоставлялась возможность вложить в продукт своего труда алгоритм сложного поведения для самостоятельного взаимодействия его творения с окружающим миром (одним из исключений является разработка идеологий существования человеческого общества).

Таким образом, создание компьютерной программы можно с большей степенью приближения уподобить акту сотворения живого организма, чем рождение, например, новой научной теории или создание нового художественного фильма. И мы имеем право сделать такой вывод в силу того, что граница между «живым» и «неживым»

определяется, в частности, и степенью сложности самостоятельного поведения «живого» и «неживого». Конечно, мы не станем утверждать, что компьютерная программа является «живой» в полном и привычном понимании этого слова, но степень ее приближения к этому состоянию и ее самостоятельная «жизнь» после «отчуждения» от автора (авторов) должна являться предметом внимательного разбора и изучения.

КОМПЬЮТЕРНАЯ ПРОГРАММА КАК ЖИВОЕ СУЩЕСТВО

Начнем с того, что, кроме поведения, компьютерная программа обладает еще одним свой-

Создание компьютерной программы можно с большей степенью приближения уподобить акту сотворения живого организма, чем рождение, например, новой научной теории или создание нового художественного фильма.

ством «живого» — способностью к размножению. Чаще всего этот процесс совершается опосредованно и при участии человека, который размножает объектный код программы при перезаписи ее с одних носителей на другие, но в случае распространения исходных текстов программы возрастает и число ее «мутаций» — видоизменений, выполненных другими программистами.

Существуют и саморазмножающиеся программы, хорошо известным примером которых является печально знаменитый класс программ-вирусов.

Большинство программ подвержены естественному отбору (при помощи человека), в котором лучшие программы заставляют «вымирать» более слабых конкурентов. При этом «выжившие» часто задают начальный уровень качества и определяют ключевые идеи, широко используемые в дальнейшем.

Многие программы обладают большими или меньшими способностями к самоорганизации (еще одной особенностью, присущей живой материи). Они могут самостоятельно упорядочивать и перепорядочивать те данные, которые накапливаются в их памяти за время взаимодействия с внешним миром и даже «отчуждать» от себя эти данные в виде отчетов, файлов, баз данных, а иногда — вместе с программами обработки этих данных — и в виде целых «организмов», способных к дальнейшему самостоятельному существованию. В этом смысле компиляторы языков программирования предназначены исключительно для анализа и перепорядочивания информации об исходной программе, написанной на одном из языков программирования, а результат их работы сводится к порождению новых выполняемых программ, обладающих собственным поведением.

Даже разработчики часто не могут детально и точно представить себе, ЧТО в данный момент происходит с их большой программной системой, или предсказать в точности ее дальнейшее поведение. Более того, если в системе используются критерии вероятностного выбора поведения системы, эта задача может быть затруднена принципиально.

Правда, при необходимости разбора и исправления ошибок в таких системах разработчики специально предусматривают механизмы поиска и локализации ошибок, включающие автоматическую запись управляющих событий и реакций системы, программную трассировку, отладочные выдачи, «посмертные» дампы и другие средства, но при этом каждый, кто хоть раз отлаживал действительно большие программы, знает, насколько нелегко порой бывает разобраться в причинах ошибок и устранить их, не внося при



этом новых ошибок. Особенно трудно это бывает в тех случаях, когда в системе широко используются параллельные процессы и/или от системы требуется работа в реальном масштабе времени.

Не зря в среде программистов бытует мудрость: «В каждой большой программе существуют как минимум три ошибки. После устранения любой из них в ней остаются или появляются еще как минимум три. И этот процесс продолжается до самого окончания жизни программы».

Из всего этого видно, насколько сложной и похожей на «живой» объект может быть компьютерная программа, которая порождается разработчиками, и насколько может быть трудно, а порой просто невозможно предсказать ее дальнейшую жизнь и поведение, особенно после «отчуждения» от создателя.

В этом месте я хотел бы проиллюстрировать вышесказанное примером длительной (десятилетней) разработки программ для большой информационной системы, базировавшейся на многомашинном комплексе. Участвуя в этом проекте в качестве разработчика, в том числе последние пять лет руководя всей разработкой и одновременно занимаясь практическим программированием, я детально представлял себе устройство всей системы в целом.

По мере возрастания числа выполняемых функций система росла и развивалась в заранее спланированных для нее рамках. Отдельные подсистемы, как это обычно бывает, проходили



различные этапы приемки, передавались сначала в опытную, а затем в промышленную эксплуатацию. Разработчиками осуществлялся авторский надзор и сопровождение, замеченные ошибки в программах устранялись, и хорошо отлаженные части системы постепенно «отчуждались» от разработчиков. Они начинали взаимодействовать друг с другом, с персоналом, эксплуатировавшим систему, и с ее пользователями, работавшими, в частности, с удаленных терминалов по линиям связи.

Надо сказать, что в период интенсивного авторского сопровождения программ я не задумывался над тем, что же это такое мы создаем с точки зрения его подобия «живому организму». Но когда стало ясно, что значительная часть системы уже «живет» своей собственной, независимой от нас жизнью и авторы уже не могут существенно влиять на эту «жизнь», не разрушая систему, я испытал известное потрясение — продукт был «отчужден» и жил самостоятельно! Причем, сохранность этой «жизни» гарантировалась заинтересованностью в нормальном функционировании системы множества других людей, пользующихся нашим «творением»... И чем больше проходило времени с момента «отчужде-

ния» системы, тем сильнее становилось ощущение самостоятельной «жизни» системы.

Думаю, что это чувство в той или иной степени знакомо почти всем программистам, продуктами труда которых пользуются другие люди. Подозреваю также, что это же чувство заставляет некоторых молодых программистов слишком гордиться своим умением написать небольшую программу и из-за этого несколько свысока относиться к окружающим.

Однако с опытом рано или поздно приходит понимание того, насколько превосходит любого из нас великий программист, сотворивший нашу Вселенную.

Но вернемся к компьютерным играм.

КОМПЬЮТЕРНАЯ ИГРА КАК ВСЕЛЕННАЯ

Действительно, разрабатывая компьютерную игру, автор определяет общие законы, по которым существует игровая вселенная, разрабатывает объекты, подчиняющиеся этим законам, наделяет их поведением, проектирует их внешний вид, их трансформации и прочее. Иными словами, создает некий мир, в котором существуют и живут сконструированные автором объекты.

Как уже говорилось, иногда этот мир может быть очень прост, жестко детерминирован и неспособен к самостоятельному существованию и развитию, а иногда — весьма сложен, подчинен вероятностным законам и способен к продолжительной жизни даже без участия играющего.

Объекты, существующие в таком мире, могут быть сконструированы и как «живые» существа, обладающие сложным поведением и самостоятельно взаимодействующие друг с другом, наделенные способностью к размножению и смерти, памятью, способностью к самообучению и многими другими свойствами.

И вновь хочу на примере собственного опыта показать, какими интересными свойствами может обладать «живая» игра.

Лет десять назад на «больших» компьютерах был очень популярен целый класс игр под названием «Star Track» («Звездный поход»). Игры с таким названием писались разными авторами и для разных компьютеров, но, в основном, под влиянием одного и того же одноименного телевизионного сериала, пользовавшегося большой популярностью в те годы. Действие всегда происходило в далеких звездных мирах, где космический корабль «Enterprise» под командованием капитана Кирка, его помощника Спока и с большой командой на борту путешествовал среди

звезд, сражался с вражеским звездным флотом («Клингонами»), защищал свои Звездные Базы, заправлялся энергией и ремонтировался, снова сражался и т.д.

Конечно, компьютерная графика в те времена была достаточно несовершенна и в лучшем случае предоставляла возможность работать с символами на алфавитно-цифровых дисплеях, но все равно «Star Track» захватывал воображение, и многие из нас ностальгически вспоминают о нем до сих пор. А чего стоили «разборы полетов» с разгоряченными после игры коллегами-программистами по дороге домой в метро! Соседи по вагону удивлялись, слыша: «...и как только я вышел из гиперпространства...»

Романтика звездных сражений настолько увлекала некоторых настоящих мужчин (программистов), что в определенный момент многие из нас начинали мечтать о своей собственной игре «Star Track», которая была бы лучше уже существующей. Не всем удавалось воплотить эту мечту в реально работающую новую программу, но многие из нас с той поры оказались «зараженными» компьютерными играми и до сих пор безуспешно ищут на персональных компьютерах свой «игровой идеал» — современную версию старой игры «Star Track».

Не устоял перед искушением написать свой собственный «Star Track» в свое время и я — и он оказался моей первой игровой программой, на которой я ощутил всю необыкновенную прелесть такой работы.

Дело в том, что эту игру я разрабатывал в первую очередь для самого себя. Все, что мне приходило в голову и хотелось видеть в готовой игре, можно было попытаться запрограммировать. Один из постоянно доступных мне компьютеров был очень неплох — HP-3000. Памяти, по тем временам, было более чем достаточно, быстродействия — тоже (к тому же я имел статус менеджера системы). Единственным ограничением, накладываемым на эту работу, являлось мое время, свободное от основной работы. Но программисту ли привыкать работать по ночам или съедать обед из термоса в выходные?

Одной из самых существенных концепций моей игры являлось вероятностное поведение объектов игры и всей вселенной в целом. Это

гарантировало невозможность точного предсказания дальнейшего развития событий в любой игровой ситуации даже для автора, что и делало игру для меня наиболее привлекательной. Программа имела специальный генерационный файл данных, содержащий около 500 параметров, относящихся к различным объектам игры.

Звезды и планетные системы зарождались и умирали, подчиняясь сложным параметрическим законам с вероятностными критериями, сверхновые взрывались в определенных ситуациях и с предварительно заданной вероятностью, оборудование космического корабля и поведение его экипажа описывалось очень большим количеством параметров и, соответственно, могло оказаться совершенно различным в сходных ситуациях. Враги, обладая очень сложной и в целом детерминированной стратегией поведения, могли отклоняться от нее в рамках разрешенной вероятности их поведения.

Каждый противник, таким образом, обладал своими склонностями и характером. Одни были агрессивны, другие — трусливы, третьи — склонны к компромиссам, переговорам и т. д. Если им удавалось договориться друг с другом, они могли в разной степени объединять свои усилия в борьбе со мной или наоборот — начать свою собственную внутреннюю войну. Враги рождались, жили и умирали по своим законам, нашу Галактику иногда посещали корабли из других галактик, команды которых могли присоединиться в зависимости от обстоятельств к той или иной враждующей стороне.

Наконец, на каком-то этапе разработки, игру стало возможно предоставлять самой себе на длительный промежуток времени, например, на ночь, и утром узнавать, к чему привело развитие событий в отсутствие автора. Если появлялось желание, можно было принять участие в игре или оставаться просто наблюдателем. (На нашем HP-3000 стояла мультизадачная операционная система с виртуальной памятью, и игра без участия человека выполнялась как задача с невысоким приоритетом, никому особенно не мешая). Примерно в это же время я решил добавить в игру возможность одновременной работы с нескольких терминалов для разных участников; и на этом этапе (примерно через год после начала работы) разработка игры собственно закончилась. Настало время наслаждаться своим

*«... и как только я вышел
из гиперпространства, смотрю —
Клингонов штук пять, а с ними
еще и Ромулан. Дай, думаю,
не снимая защиты, врежу им всем
фотонными торпедами,
а надо-то было сразу бить
фэйзером — близко они стояли...
Потом еле до Базы добрался,
но только вошел в док —
атаковали и Базу...»*

«творением» и иногда отдыхать за дисплеем компьютера; можно было подвести некоторые итоги.

Но именно в этот момент у меня появился первый персональный компьютер Yamaha MSX, обладавший цветной графикой и захвативший на длительное время все мои помыслы.

Итоги разработки «Star Track» были подведены мной значительно позже и, честно говоря, обрадовали и даже немного испугали одновременно. Я был рад, что написал такую большую и сложную программу и выполнил задачу, которую перед собой поставил в полном объеме. Я гордился тем, что в очередной раз создал систему, способную к дальнейшему существованию без участия автора, и что моей игровой программой с удовольствием пользовались некоторые мои коллеги по работе. Но одновременно с этим я понял также и то, что на этот раз соприкоснулся с чем-то большим, чем обычное проектирование компьютерной программы. Некоторый неожиданный опыт заставил меня сделать некоторые выводы, не имеющие прямого отношения к программированию в общепринятом понимании этого слова.

Во-первых, мне довелось на практике убедиться в справедливости концепции, носящей название «Лучший из миров». Суть этой философской и физической концепции, как известно, заключается в том, что мы живем в «лучшем из миров» в том смысле, что наша Вселенная является устойчивой в силу вполне определенного соотношения значений таких физических констант, как скорость света в пустоте, постоянная Планка, заряд электрона и др. Соответствующими расчетами показано, что существует лишь очень небольшая область изменений мировых констант, обеспечивающая множеству возможность оставаться устойчивыми. И наша Вселенная принадлежит к этому небольшому множеству. В случае больших отклонений значений этих констант в ту или иную сторону материя начинает стремительно дезорганизовываться и превращается в хаос или, что то же самое, не может организоваться из первичного хаоса вовсе.

Как я уже говорил, моя игра имела около пятисот параметров, определяющих ее игровую вселенную, и правила поведения объектов, включая вероятностные характеристики отклонений параметров от их стандартных значений. В процессе проектирования и настройки игры с помощью этих параметров выяснилось, что возможные устойчивые игровые вселенные также занимают очень небольшой объем в многомерной области допустимых значений всех параметров. А при выходе одного или нескольких параметров за границы этого объема игровая вселенная начинает стремительно деградировать и становится однородной и неинтересной для

играющего. Например, сверхновые звезды вспыхивают слишком часто и сжигают все вокруг себя, новые звездные системы зарождаются слишком редко или слишком часто, персонажи самоуничтожаются, вымирают или уничтожают друг друга в братоубийственных войнах, оборудование корабля становится слишком ненадежно, команда корабля постоянно бунтует и плохо выполняет приказания или, наоборот, проявляет излишнюю самостоятельность и делает то, что делать совсем не надо... Перечисление можно легко продолжить. Таким образом, понятие «Лучший из миров» для данной игры приобрело сугубо практическое значение, и мне пришлось выбирать этот «Лучший из миров» аналитически, интуитивно, а также методом проб и ошибок в процессе настройки программы. Более того, стало особенно важно попасть в один из «Самых лучших миров» для того, чтобы игровая вселенная могла самостоятельно существовать без моего участия.

Во-вторых, постепенно развивая игру и надевая ее персонажей характерами с различными вероятностными характеристиками, я вовсе не думал о том, что, вообще-то, ничто не мешало мне запрограммировать у них интерес к познанию окружающего их мира и даже способность к рефлексии. Те «существа», которых я создавал, уже обладали некоторой, независимой от меня индивидуальностью. Они рождались и умирали помимо моего желания, имели свои собственные имена, которых я не знал до тех пор, пока незнакомился с каждым из них в отдельности, обладали достаточно сложным поведением, в большей или меньшей степени адекватным окружающему их миру, некоторым небольшим запасом слов и понятий (я с ними беседовал и иногда мог даже «убедить» в чем-то), они объединялись в группы с общими интересами или, наоборот, враждовали друг с другом. В моей модели не хватало совсем немного: их собственной заинтересованности в продолжении игры и осмысления ими в этой игре своей роли. Кроме того, хотя они и «знали» о моем существовании, но не могли целенаправленно задавать мне вопросы о том мире, в котором существуют, и получать от меня соответствующие ответы.

Когда я впервые задумался над этим, я уже, к сожалению, не имел доступа к тому компьютеру, на котором создавал эту игру. Поэтому я не имел и возможности попытаться дополнительно запрограммировать эти черты их «личности» и посмотреть, что же из этого получится (а на сегодняшний день, работая с персональными компьютерами, я вынужден почти всю их память использовать под графику и в первую очередь думать о красивом и правдоподобном ото-

бражении информации на экране в ущерб программированию «личностных» характеристик персонажей игр, ведь иначе на эту игру никто и смотреть не захочет). Однако, чисто теоретически, эта задача не представляется мне особенно сложной, так как еще с очень давних времен у меня уже был опыт разработки обучаемых программ, синтезирующих связный русский текст с расширяемым в процессе обучения запасом понятий и оперирующих различными категориями (в том числе морально-этическими). А развитый язык понятий — один из основных критериев степени развития интеллекта и знаний об окружающем мире!

Дальнейшие размышления по поводу того, во что мог бы превратиться мой «Star Track», привели меня к некоторым неожиданным религиозно-философским аналогиям. Я задавал себе вопросы и пытался отвечать на них. Итак:

- Зачем я создавал эту программу? Отчасти ради самого удовольствия от процесса ее создания и предвкушения удовольствия от дальнейшей «игры» с ней. Отчасти для расширения разнообразия окружающего меня мира. Отчасти для удовлетворения чувства созидания, живущего в каждом человеке. Отчасти просто из любопытства. И еще мне хотелось иметь возможность исследовать в деталях все время новый, постоянно обновляющийся мир игры, сталкиваясь порой с неожиданными явлениями в рамках предварительно определенных мной законов этого мира.

- Почему я вынужден был искать для этой игры свой «Лучший из миров»? Потому, что вырожденная игра, хаос мне **неинтересны**. Мне **нравится** уменьшать энтропию, идти от хаоса к упорядоченности.

- Зачем «живут» мои персонажи? Для того, чтобы, изменяя в бесконечных сочетаниях и вариациях свой мир, доставлять мне постоянную радость исследования и познания этого изменяющегося мира.

- Как я отношусь к самоуничтожению своих персонажей? Резко отрицательно. Любое «самоубийство» персонажа, с моей точки зрения, есть невыполнение им возложенных мной на него надежд на участие в игре.

- Как я отношусь к персонажам своей игры? Ведь среди них есть «плохие» и «хорошие», «добрые» и «злые», «умные» и «глупые»! Конечно, субъективно, как участнику игры, мне приятней иметь дело с «хорошими», «добрыми» и «умными» персонажами, но без «плохих» игра выродилась бы и стала бы мне **неинтересна**. Объективно я могу сказать, что мне необходимы все персонажи и что я, как творец, одинаково **люблю** их **всех**, хотя, безусловно, больше сочувствую «хо-

рошим». Необходимость во **всех** персонажах становится особенно заметной в том случае, когда я лично не участвую в игре и выступаю лишь в качестве наблюдателя.

- Как поведет себя мой персонаж, если в результате рефлексии вдруг придет к выводу, что он является «существом», а я являюсь творцом его мира? Предположительно, он захочет задать мне вопросы и получить на них ответы. Но большая часть моих ответов в той или иной степени будет соотноситься с **моим** миром, о котором этот персонаж не имеет **никакого** представления! Какая бездна новых знаний и понятий! Захочет ли он после этого возвращаться в свой простой игровой мир или будет все время стремиться «медитировать», общаясь со мной? Интересно и то, что большая часть знаний, которые я мог бы ему сообщить, оказалась бы совершенно неприменимой в рамках его вселенной.

- Как поступать с машинной памятью и хранящимися в ней упорядоченными за время «жизни» персонажа данными, остающимися после его «смерти»? В этом месте каждый программист, читающий эту статью, может сам поразмыслить на эту тему и предложить наиболее соответствующее его пониманию проблемы и вкусам решение.

Могу добавить, что разработка и анализ этой игры в свое время так сильно повлияли на мое мировоззрение, как не смогли повлиять до этого никакие теоретические концепции. Согласитесь, одно дело — рассуждать о чем-то схоластически, а другое дело — попробовать самому.

Здесь можно было бы поставить точку. Но просится многозначительное многоточие...

О ТЕХНОЛОГИИ РАЗРАБОТКИ ИГРОВЫХ ПРОГРАММ

Теперь от «большой игры» на «большой машине» перейдем к рассмотрению технологий разработки игр для современных персональных компьютеров.

Как известно, процесс создания таких игр состоит из четырех составляющих:

- написание сценария;
- разработка графики игры;
- написание музыкального сопровождения;
- разработка компьютерной программы, объединяющей все три вышеперечисленные компоненты в единое целое, называемое компьютерной игрой.

Для того чтобы процесс создания игры средней сложности не растянулся на несколько лет и

конечный продукт был бы получен в короткие (или просто разумные) сроки, необходимо иметь определенный набор инструментов, который я в дальнейшем для краткости буду называть просто «верстак».

Если таких инструментов нет или их не хватает, то, прежде чем приступить к разработке игры, лучше всего создать эти инструменты самому. Ведь вы не станете, к примеру, делать табуретку, если у вас нет рубанка! Конечно, можно попытаться выстругать детали табуретки перочинным ножом или топором, но какая табуретка при этом получится и сколько времени это займет? А сколько новых табуреток можно было бы изготовить за то время, пока делали эту?

Итак, что же должно быть на вашем «верстаке»?

1. Удобный редактор текстов для написания сценария, подготовки документации и разработки исходных текстов программ.

2. Мощный графический редактор для подготовки фоновых задних планов и анимационных объектов. Если планируется использование трехмерной графики, может понадобиться и 3D-редактор. Неплохо также иметь сканер, видеокамеру и устройство ввода в компьютер видеoinформации. Форматы хранения изображений в файлах на дисках должны поддерживаться удобными и простыми библиотечными функциями для последующей загрузки графики в вашу программу. Если вы планируете игру для CD-ROM, может быть, вам придется снимать целые фрагменты видеофильмов, чтобы потом показывать их с помощью вашей программы.

3. Удобный музыкальный редактор, работающий с той музыкальной аппаратурой, которую вы рассчитываете использовать в своей игре. Все следует заранее определить. Если это обыкновенная «пищалка» IBM PC, то ее звуковые возможности невелики, хотя она позволяет, пусть очень тихо, но воспроизводить оцифрованную музыку. Если это музыкальный синтезатор (встроенный — как в MSX или Atari ST, или подключаемый в виде дополнительной карты — как в IBM), то ваши возможности богаче. Если же это специальные встроенные каналы вывода цифровой информации в звуковом виде (как в компьютерах Amiga, Macintosh, NeXT), значит вы можете создавать в своей игре «настоящую» музыку и речь, а также использовать записанную в других музыкальных студиях. Так же, как и для графиче-

ки, вам должны быть доступны соответствующие библиотечные функции для удобного проигрывания созданной музыки из вашей программы.

Хорошо, когда музыка, используемая в игре, мелодична и удачно аранжирована. Поэтому, если вы не уверены в собственных силах, попросите написать музыку для вашей игры профессионального музыканта. В противном случае ваша игра может потерять привлекательность, даже при интересном сценарии и хорошо продуманном алгоритме. То же самое, впрочем, относится и к графике.

4. Необходим компилятор с языка программирования, на котором вам удобнее всего излагать свои мысли, и ассемблер. С точки зрения компактности письма и обзорности текста программы лучше всего писать и отлаживать на

языках высокого уровня, но, если эффективность порождаемого компилятором кода оказывается недостаточной, отдельные (наиболее критичные по быстродействию или использованию памяти) функции приходится переписывать на ассемблере. Очень удобно пользоваться объектно-ориентированными языками, но, к сожалению, компиляторы с таких языков пока недостаточно широко доступны всем желающим на различных компьютерах.

Для себя, например, я сделал выбор языка программирования несколько лет назад, остановившись на языке Си. Перед этим я писал программы на более чем десятке разных языков и ассемблеров на различных компьютерах, сам разработал и реализовал несколько специализированных языков и макропроцессоров, писал даже непосредственно в двоичных кодах и абсолютных адресах (первый раз еще в школе), но, познакомившись с Си, понял, что в терминах этого языка мне удобнее всего думать и излагать свои мысли.

Конечно, и Си не лишен недостатков. Я, например, знаю и такие языки, где знак умножения и значение переменной типа «указатель» обозначаются разными символами (умножение '*', а указатель '@'), но сегодня, если меня лишит удовольствие описывать конструкции типа

```
if (! (i == (++j > k)) ) puts("Damned Zero again!");
```

я буду испытывать определенный дискомфорт.

5. Вам могут потребоваться отдельные утилиты, осуществляющие сжатие/расширение ин-

*Хороший «верстак» всегда
предохранит вас от массы мелких
и досадных ошибок, так как
позволяет максимально компактно
описывать алгоритм работы
вашей программы.*

формации, кодирование областей на дисках для защиты от копирования ваших программ, средства работы с библиотеками и автодокументатор для исходных текстов или любые другие в зависимости от необходимости и конкретных потребностей каждого разработчика.

6. И, наконец, может быть, самое важное. Вам нужна библиотека функций, с помощью которой можно легко и удобно работать с графическими и звуковыми данными, опрашивать все необходимые вводные устройства (мышь, джойстик, клавиатуру, специальные контроллеры), осуществлять анимацию объектов и управлять ими, следуя логике вашей игры и не отвлекаясь на «заклинания», необходимые для взаимодействия с компьютером через различные системные интерфейсы, требующие выполнения множества системных соглашений.

Проблема «заклинаний» иногда приобретает просто чудовищный оттенок в некоторых мультитасковых средах, где большинство системных ресурсов, к которым необходимо получить доступ,

требуется сначала описать, потом открыть, заполнить предварительно подготовленные соответствующие структуры данных, и, наконец, поработав с этими ресурсами (далеко не всегда удобными для вас лично примитивами системных операций), освободить, закрыть и вернуть системе.

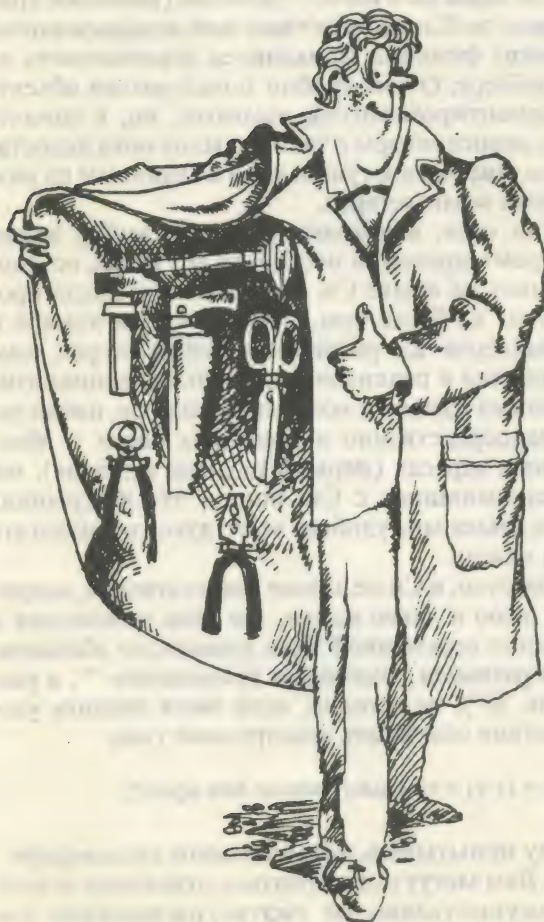
Если вы не будете делать всего этого, вам придется довольно часто перезагружать систему, а ваша новая программа, скорее всего, не будет работать на других, таких же, как у вас, компьютерах, но всего лишь иначе сконфигурированных. Порой, пока вы планируете и программируете доступ к этим ресурсам, можно вообще забыть, зачем вы сели за компьютер и какую задачу собирались решать.

Единственным корректным выходом из этой ситуации является погружение всех «заклинаний» в ваши собственные библиотечные функции и макросы при условии, конечно, что они понадобятся вам и в дальнейшем.

Но это лишь одна из причин, по которой разработку каждого нового класса программ лучше всего начинать с проектирования или расширения вашей собственной библиотеки.

Другая (и более важная) причина заключается в том, что, если вы хотите действительно **быстро** писать программы данного класса (в частности, игры), не отвлекаясь на второстепенные детали, вам необходимо определить, в каких терминах и операциях вам удобнее всего описывать алгоритмы работы программ этого класса (включая языковые конструкции используемого вами языка программирования). Другими словами, следует понять, каким вам хотелось бы видеть язык, на котором вы будете писать свою программу. Это очень важная работа, ее желательно проделать в начале, т.е. перед тем, как написать самую первую строку вашей самой первой программы любого нового для вас класса (как уже было замечено, здесь могут оказать большую помощь объектно-ориентированные языки программирования).

Результатом этой работы должно стать множество структур данных и операций, минимизированное по количеству функций и их параметров, что позволит наиболее компактно описать алгоритм работы вашей программы. Это очень тонкая работа, требующая известного опыта и являющаяся своего рода искусством: исходя из общей постановки задачи (например, в виде сценария) и знания архитектуры вашего компьютера (или архитектур всех известных вам компьютеров), следует придумать максимально эффективные примитивы операций, с помощью которых вы могли бы **предельно компактно** описать все ваши алгоритмы и которые не снижали бы



существенно эффективности работы ваших готовых программ!

Сверхзадачей при этом является максимально возможная мобильность (переносимость на компьютеры с другой архитектурой) ваших программ за счет того, что вся конкретная работа с аппаратурой каждого отдельно взятого компьютера может быть полностью описана вашими новыми примитивами. Таким образом, один раз переписав ваши примитивы для компьютера с другой архитектурой и пользуясь родственным компилятором того же языка программирования, вы сможете в дальнейшем легко переносить ваши программы данного класса на другие компьютеры.

При внимательном рассмотрении данной концепции легко заметить, что множество примитивов, о которых шла речь выше, вместе с конструкциями используемого языка программирования образуют некоторый новый язык, на котором и описывается алгоритм работы программы. Часть примитивов погружается в библиотеки, а часть выносится на уровень макросов языка программирования. При этом, как показывает практика, на уровень макросов чаще всего выносятся узкоспециализированные обращения к тем же библиотечным функциям или их использование в различных частных комбинациях, увеличивающие выразительную силу нового языка.

Правда, такой подход несколько противоречит концепции структурного программирования Н. Вирта, в которой любая разработка должна проектироваться догматически строго сверху вниз, но, во-первых, Вирт и его последователи замалчивали этап предварительной разработки языка под задачу, во-вторых, в программах, работающих в реальном масштабе времени, всегда важно знать и оценивать реакции аппаратуры на различные сложные воздействия, а, не опробовав каждую новую программную функцию нижнего уровня, такие оценки давать очень трудно. К тому же разработка большинства программ идет почти всегда сразу в двух направлениях: от аппаратуры — снизу вверх и от логики разработки всего проекта — сверху вниз. Интересно, что после «обкатки» каждого нового набора примитивов на нескольких программах данного класса становится возможным и даже удобным пользоваться в дальнейшем исключительно структурным подходом при проектировании новых программ данного класса.

Таким образом, ваш «верстак» должен содержать немалое количество хороших специальных инструментов, подходящих для вашей работы. Но если вы не пожелаете времени на сбор этих инструментов, их освоение и разработку недостающих, то скоро у вас появится возможность «печь» ваши

программы как блины — быстро и качественно!

На своем опыте я уже не раз убеждался в этом. Потратив известное время на изготовление «верстака» для совершенно «голого» в те времена компьютера MSX (1986—1987 гг.), я затем всего за полгода написал четыре(!) игровых программы выше средней степени сложности. Причем каждая из них разрабатывалась с нуля, включая сценарий, музыку, графику и собственно игровую программу. Таким образом, на каждую игру от начала разработки (первого слова сценария) до ее окончательного завершения (вместе с музыкой и графикой) уходило всего полтора месяца работы одного человека!

Все четыре игры относились к классу «приключенческих» игр (Adventure), обладали сложными сценариями, развитой анимацией, экранными скроллингами больших полиэкранов, специальной многооконной организацией графической информации и генерировали свои игровые вселенные с учетом вероятностного распределения параметров, что исключало возможность повторения одних и тех же игровых ситуаций. Каждый сеанс игры был новым приключением с новыми отличиями в каждой конкретной игре.

Этот подход я использую и сейчас, в том числе при разработке программ для компьютеров с другой архитектурой. Помимо всего прочего, хороший «верстак» всегда предохранит вас от множества мелких и досадных ошибок, так как позволяет максимально компактно описывать алгоритм работы вашей программы.

Но, углубившись в пучину технологии, не забывайте, пожалуйста, что игры бывают «добрые» и «злые», а также, что, согласно Словарию русского языка (под ред. А.П. Евгеньевой), одно из определений игры — «свойственное некоторым винам и шипучим напиткам движение пузырьков газа».

ЗАКЛЮЧЕНИЕ

Я надеюсь, что если хотя бы один читатель этой статьи узнает из нее что-нибудь новое или полезное для себя лично, то моя задача уже выполнена.

А если какой-нибудь мой коллега-программист прольет слезу умиления над этой статьей, потому что сам давным-давно пришел к тем же выводам, я буду просто счастлив.

Во всех остальных случаях не судите меня слишком строго, ведь игры — дело совершенно несерьезное. В них играют маленькие дети. А большие дети их разрабатывают. ♦



Игры приносят успех? Да, если повезет.

Фирма «ПараГраф» известна в мире своей технологией распознавания слитного рукописного текста. Но ее деятельность не ограничивается только этим направлением, здесь ведутся работы над распознаванием голоса, разрабатываются прикладные программы для машин с рукописным вводом и т.д. Много усилий и средств вкладывается в разработку компьютерных игр. О компьютерных играх и состоянии их рынка редакторы журнала «Мир ПК» Эдуард Пройдаков и Анна Павловская беседуют с директором отдела игр этой фирмы Георгием Пачиковым.

— Каким фирмам принадлежит главенствующая роль на западном рынке игровых программ? На кого в основном ориентирована их продукция?

— В количественном отношении на рынке компьютерных игр преобладают игры для телевизионных приставок и карманных устройств типа Game Boy, выпускаемых в основном фирмами Nintendo, SEGA и NEC. Цена на приставки и устройства невысокая. Она колеблется где-то в пределах 100 — 150 долл., и почти в каждой американской семье есть такое электронное устройство. Компаниям, производящим компьютерные игры, очень выгодно создавать игры именно для этих устройств, так как каждый их обладатель является потенциальным покупателем новых игр. Поэтому игры для телевизионных приставок и устройств типа Game Boy разрабатывают буквально все известные игровые компании: Electronic Arts, Activision, Spectrum HoloByte, Broderbund, Taito. Разумеется, рынок компьютерных игр этим не ограничен. Игры для персональных компьютеров тоже создаются, но их гораздо меньше.

Если говорить о потребителях, то это в основном подростки 10—15 лет. Именно на мальчишек, которые могут, заработав или попросив немного денег у родителей, купить понравившуюся игру, ориентируются многие компании, работающие в области игрового бизнеса. А что в таком возрасте мальчишек интересует? Стрельба, бои, приключения. Такими играми и завален западный рынок. Они отличаются друг от друга качеством графического интерфейса, но с точки зрения игры как некоторого набора правил — удивительно однообразны. Я называю их «играми для рук», потому что, играя в них, не надо думать, нужно только быстро двигать джойстик. Однако подростки предпочитают именно такие военные, изобилующие насилием игры. И на Западе это воспринимается как проблема!

— Как Вы относитесь к «заигрыванию», к уходу детей в искусственную реальность, то, что называют также «компьютерной наркоманией»? Интересно, сталкиваетесь ли Вы с этой проблемой в детском клубе, над которым шефствует Ваша фирма?

— Проблема состоит несколько в другом. Детей провоцируют играть в однообразные и, на мой взгляд, тупые игры. И в этом смысле это, конечно, компьютерная наркомания. В нашем детском клубе проблема несколько другая. Здесь некоторые дети настолько увлекаются программированием, что тратят на это почти все свое время. Они, можно сказать, живут в компьютере, забывая о мире вне этой «железки». Поэтому мы изменили направленность деятельности клуба. Мы видим теперь цель не в том, чтобы из всех занимающихся здесь детей воспитать программистов, а стараемся так построить занятия, чтобы дети поняли, что компьютер — это прежде всего средство общения, и научились использовать его для общения со своими ровесниками во всем мире. А общение, я уверен, вызывает потребность в знаниях. Если же говорить об играх, то у нас можно играть лишь в игры, разработанные внутри клуба.

Недавно я с удивлением узнал, что вице-президент фирмы ATARI Лари Сигал запретил своим детям играть в компьютерные игры, мотивируя это заботой об их развитии. Я не столь категоричен. Есть много полезных игр. Моя дочь уже в 3—4 года играла на компьютере в Puzzle (это игра, в которой нужно собрать картинку из фрагментов).

— Какие направления в компьютерных играх для детей, на ваш взгляд, наиболее интересны? Есть ли на Западе альтернатива военным играм?

— Не вдаваясь в тонкости и детали, все детские компьютерные игры можно подразделить на «игры для казино» и исследовательские. И, безусловно, исследовательские игры для детей — это альтернатива бездумным военным играм. В этом направлении работает фирма Broderbund, специализирующаяся на компьютерных играх и образовательных программах. Она выпустила серию игр, которые одновременно являются и образовательными, и развлекательными. Главный герой этой серии — женщина-шпион Кармен Сандiego оказывается в разных ситуациях в разных местах земного шара. Для того чтобы поймать Кармен, необходимы знания по географии. Эта игра, в частности, вызывает в детях интерес к географии. Вообще, этот стык очень сложный — трудно сделать игру одновременно и интересной, и познавательной, но Broderbund это удалось.

В последнее время эта фирма перешла к созданию игр нового типа, рассчитанных на детей 5—8 лет. Это так называемые живые книжки (Living book). Основу Living book составляют

классические сказки. Такие игры дают детям возможность играть в сказку и, играя, исследовать ее мир, который разворачивается на экране компьютера. Ткнув, например в автобус, «мышью», можно его оживить и поехать с ним в другую картинку. Можно спросить героя про домик, изображенный на экране, и герой вам расскажет про него. Это очень перспективное направление, Living book хорошо раскупаются. Интересный момент: многие бизнесмены стали создавать фирмы, основным капиталом которых является купленная лицензия на какую-нибудь известную детскую книжку. Но надо сказать, что игры, разрабатываемые в рамках этих двух направлений, ориентированы не на телевизионные приставки. Создание Living book стало возможным благодаря тому, что разработчик получил в свое распоряжение мощные персональные компьютеры с прекрасным звуком и палитрой и практически неограниченным размером оперативной памяти.

— Но за компьютером играют не только дети. Взрослым это занятие тоже нравится, не так ли?

— Конечно, но, как правило, взрослые играют в карточные игры. И я думаю, что они для них — больше, чем игра. Для меня, например, компьютерный пасьянс — это отдых, возможность подумать о чем-то своем. А игра — просто фон. Разрабатывая для них игры, надо понимать, что у взрослых нет времени, и поэтому игра для взрослого человека должна быть такой, чтобы он быстро «вошел» в нее, поиграл и быстро вышел, т.е. в любой момент мог бы прекратить играть.

Все это касается традиционных, обычно хорошо продаваемых игр. Но есть другое очень интересное направление в играх, которое, как мне кажется, получит развитие в ближайшем будущем. Я имею в виду поведенческие игры. Одна из них называется Кевин (по имени ее главного героя). Кевин — это человечек со своим характером, своей жизнью. Его можно о чем-нибудь попросить, например сыграть на пианино. Если у него хорошее настроение, он вам сыграет. Он также может обратиться к вам с какой-нибудь просьбой. Герой, живущий в игре, в каком-то смысле может составить компанию одинокому человеку. Почему мне кажется, что будущее за этими играми? Завтра, когда будет создана виртуальная реальность, с вами будет общаться в игре, будет сострадать вам «живой человек». Но общение подразумевает двустороннюю связь, т.е. нужна ответная реакция. Ее реализация — основная проблема в этих играх. Вообще говоря, понятно, как это можно было бы сделать: чело-

век, надев на палец датчик, будет «сообщать» игре о своем состоянии, а игра будет соответственно модифицироваться. И конечно, чтобы создать такую игру, нужна помощь психологов. Такие программы могут найти применение в медицине, в частности, можно будет помочь людям, находящимся в стрессовой ситуации.

Еще одно очень интересное направление — психологические игры (не надо путать их с психологическими компьютерными тестами). Очень интересные трехмерные лабиринтные игры делает Володя Пахилько. Он работает вместе с А. Пажитновым в фирме BPS. Когда наше подразделение только начинало свою деятельность, мы тоже пытались сделать психологическую игру, но столкнулись с непониманием со стороны психологов. Хотя мне по-прежнему кажется, что игра, оценивающая человека, будет иметь свой рынок. Основная проблема при разработке таких игр в том, как во время игры получить информацию, чтобы оценить человека. Очевидно, его нужно провести в ней по таким ситуациям, реакции на которые позволили бы делать соответствующие выводы.

— Можно ли говорить о рынке компьютерных игр в нашей стране?

— Пока такого рынка нет. Можно говорить о рынке производителей компьютерных игр, и следует заметить, его активно используют западные бизнесмены. Имея какое-то представление о том, что можно продать на западном рынке компьютерных игр, они приезжают сюда и нанимают программистов, чтобы те написали нужные им игры. В качестве гонорара дают компьютер. Представляете, что такое для нашего программиста получить компьютер? Многие соглашаются работать на таких условиях, чтобы приобрести инструмент для разработки новых игр.

— В странах СНГ распространены в основном машины класса AT 286. Поскольку в США эти машины — вчерашний день, разработка игр для них чревата тем, что эти игры не пойдут на западном рынке из-за недоиспользования производительности новых машин. Что Вы об этом думаете?

— С моей точки зрения, главное в разработке игр — придумать правила игры, а не ее реализация. Если наше подразделение возьмет новую, не очень сложную игру, то мы ее сделаем в течение месяца. Имея инструментальный (а любая фирма, создающая игры, имеет необходимый инструментальный — анимационные пакеты, звуковые и

графические пакеты, рисовальные пакеты, набор библиотек и т.д.), можно очень быстро создать игру. Если компания опытная, то реализовать сценарий несложно. С этим связана одна очень серьезная проблема для наших разработчиков. Предлагая продавать сделанную вами игру, например, в США, западный бизнесмен, взяв вашу идею, может нанять двух-трех программистов или найти какую-нибудь компанию, которая буквально в течение 2—3 недель сделает игру, частично изменив исходный вариант. Иначе говоря, передавая свою игру или даже просто показывая ее, разработчики всегда рискуют.

— Каким же образом можно застраховать себя от таких ситуаций?

— Я не знаю способа, как избежать этого риска. Видимо, в условиях контракта необходимо оговаривать первоначальные выплаты. Как поступают западные бизнесмены? Они берут игру у наших разработчиков и якобы начинают ее продавать. Понятно, что они ничем не рискуют, занимаясь ее продажей между делом, ничего в это не вкладывая. Если бы они что-нибудь вложили, они попытались бы вернуть свои деньги, причем чем больше они вложат в игру, тем с большей отдачей будут работать на западном рынке. Это самая большая проблема, с которой приходится сталкиваться нашим программистам. Нельзя отдавать исключительные права на игру, рассчитывая на проценты от будущих продаж. Их может и не быть.

— Интересно, как велики эти проценты?

— Для телевизионных приставок отчисления разработчику за игру обычно составляют 2—3% за один картридж, а если очень повезет, то 4—5%. Для персональных компьютеров эта величина колеблется от 7 до 15%. Это зависит от издательских расходов. Кстати, с этим связан один интересный момент — коробки. Как говорят западные издатели, если покупаешь товар, он должен хорошо выглядеть. Человеку психологически легче купить игру, если она упакована в большую, красивую коробку, пусть там и лежат только одна дискета и книжечка-инструкция.

— Что бы Вы хотели сказать нашим читателям в заключение?

— Разработчики игр призваны создавать новые игры, а не повторять старые идеи, когда просто берется какая-то известная игра, меняются персонажи и «фон», на котором происходит действие, и то, что получилось, выдается на рын-

ке как новая игра. Очень многие программисты потратили массу времени на разработку игр типа TETRIS и PUZZLE. Потом они жутко разочаровывались, узнав, что таких игр очень много. Это типичная проблема общества, где люди работают, не зная рынка.

Многие программисты думают: чтобы достичь успеха — проще всего написать игру. Я это называю феноменом Пажитного. TETRIS Пажитного имеет необычный, колоссальный успех. Это самая популярная в мире игра. Она идет практически на всех типах машин, начиная от маленьких игровых компьютеров фирмы

Nintendo, которые просто продаются с этой игрой, и кончая рабочими станциями фирмы SUN. Но такая игра, как TETRIS, появляется очень редко, один раз в столетие, наверное. Конечно же, это гениальная игра, ее успех во многом объясняется тем, что она появилась в то время, когда Запад переживал огромный интерес к Советскому Союзу.

И еще. Надо понимать, что рынок компьютерных игр, пожалуй, самый сложный. Требуются огромные затраты на рекламу, на поиск потенциальных потребителей. Да, это самый успешный рынок, но при одном условии: если повезет.

НОВОСТИ

Коммерческие глобальные сети

24—25 декабря 1992 г. в Санкт-Петербурге проходил семинар по коммерческому использованию глобальных компьютерных сетей, организованный АО ICE (Москва) и Институтом ТЕЛЕКОМ (Санкт-Петербург).

В работе семинара приняли участие многие специалисты в области телекоммуникаций и банковской деятельности. В своем вступительном слове председатель Комитета мэрии Санкт-Петербурга по связи, информатике и транспорту В.Т.Шилягов отметил необходимость стандартизации банковских систем, без которой невозможен переход к единой электронной системе взаиморасчетов. Для координа-

ции деятельности банков по развитию сетевых технологий планируется организовать в Москве и Санкт-Петербурге серию семинаров. На семинаре демонстрировалась работа коммерческой сети передачи данных и документального обмена ИСТОК-К (построенной на базе технических средств ведомственной сети ИСТОК) и информационной сети PIE-Net. По оценкам специалистов, несмотря на неразвитую инфраструктуру связи и многочисленные трудности в ее организации, уже сегодня коммерческое использование глобальных компьютерных сетей может обеспечить неплохую отдачу. Контактный телефон Института ТЕЛЕКОМ: (812) 560-06-30

Новый проект National Instruments

Фирма National Instruments является признанным лидером в сфере производства аппаратного и программного обеспечения для интерфейса GPIB*. Любое заинтересованное лицо может принять участие в объявленном ею проекте. Ему будет предоставлена интерфейсная карта GPIB для IBM-совместимого ПК, карта сбора данных (plug-in data acquisition card) и пакет LabWindows. В обмен на это участник проекта обязан разработать и передать компании

* Аналог приборного интерфейса IEEE-488 (КОП, канал общего пользования). — (Прим. ред.)

драйверы двух подключаемых к каналам GPIB, VXI или RS232 приборов западного производства.

Каждый желающий принять участие в проекте должен располагать IBM-совместимым компьютером, иметь доступ к двум приборам западного производства и уметь программировать на языке Си.

Свяжитесь с нами или посетите наш стенд на выставке MERA, которая будет проходить в Москве с 25 по 31 марта 1993 г.

Aram Salatian
National Instruments
Finland OY
ATT: Aram Salatian
PL2
02631 ESPOO, Finland
Tel. (358) (0) 527-23-21
Fax (358) (0) 502-29-30

Семинар Intel

Корпорация Intel провела в Москве 8—9 декабря прошлого года семинар для своих дистрибуторов. Выступая на семинаре, региональный менеджер по продажам Стивен Чейз отметил, что особое внимание корпорация уделяет сегодня средствам расширения возможностей персональных компьютеров — сопроцессорам, процессорам-ускорителям OverDrive, модемам, факс-модемам, сетевому оборудова-

нию и программному обеспечению для сетей. Результаты деятельности в этом направлении налицо: по поставкам сетевых адаптеров за последние 12 месяцев корпорация вышла на третье место в Европе. Вниманию участников семинара был предложен недавно разработанный Intel индекс определения производительности микропроцессоров iCOMP (Intel Comparative Microprocessor Performance index). Необходимость в подобном индексе обусловлена тем, что неко-

торые традиционные характеристики, например, тактовая частота, могут вызывать у потребителя неверное представление о быстродействии микропроцессора. Например, индекс iCOMP для 33-МГц микропроцессора 386DX равен 68, а для 20-МГц 486SX — 78 несмотря на более низкую тактовую частоту. Индекс iCOMP представляет собой среднее взвешенное результатов различных тестов, и его формула открыта для всех заинтересованных компаний.

В ходе семинара была также объявлена программа iANR (Intel Advanced Network Reseller) создания сети фирм-дилеров для продажи сетевой продукции Intel. В поисках потенциальных партнеров Intel ориентируется в первую очередь на те фирмы, которые уже занимают прочные позиции в соответствующем сегменте рынка. Корпорация Intel готова предоставить своим партнерам iANR приоритет в получении информации о новой продукции, техническую поддержку и льготные цены.

Из жизни логических игр: IV Международная компьютерная олимпиада

И.В. Григорьев

В начале августа 1992 г., когда на спортивных аренах Барселоны определялись сильнейшие, дух олимпийских баталий ощущался не только в Испании — в Лондоне на Пиккадилли залы пятизвездного отеля «Парк Лейн» принимали своих олимпийцев. Эти люди не претендовали на звания самых сильных и ловких, и вообще за олимпийское золото сражались не они, а их программы. Дело в том, что с 5 по 11 августа 1992 г. здесь проходила IV Международная компьютерная олимпиада.

Все началось с пари. В 1968 г. Дэвид Леви — международный шахматный гроссмейстер, загоревшись идеей проведения соревнований между компьютерными программами по наиболее известным видам логических игр и приложив огромные усилия для ее воплощения, заключил пари со специалистами в области искусственного интеллекта на 1250 английских фунтов стерлингов, что в течение 10 лет ни одна шахматная компьютерная программа не сможет его победить. Он выиграл пари и продолжал оставаться непобедимым для компьютера еще 11

лет. Однако компьютер, это стремительно растущее дитя научно-технического прогресса, развивает свои возможности не по дням, а по часам. С 1970 г. ежегодно проводится по несколько шахматных турниров, в которых за шахматными досками «сидят» компьютеры. Однако шахматы являются далеко не единственной логической игрой, привлекающей интерес программистов. Существует много других игр (шашки, реверси, нарды, го, го-моку, бридж), раскрывающих «интеллект» компьютера практически во всех его ипостасях. Сегодня число программ, умею-

щих играть в подобные игры, так велико, что, похоже, наступило время для проведения между ними состязания мирового масштаба. И если в 1896 г. барон Пьер де Кубертен дал второе рождение Олимпийским играм, то в 1989 г. Дэвид Леви дал старт новому олимпийскому движению, соответствующему реалиям нашего времени.

Правила турниров компьютерной олимпиады аналогичны правилам проведения состязания между людьми за тем исключением, что противниками за игровой доской являются компьютеры, люди же (операторы) только осуществляют контакт между ними. Соревнования проводятся по различным «видам спорта». В 1992 г. в олимпийский список входили: Авари (аналог Каллаха), Нарды, Отелло (Реверси), Го (9×9 клеток), Го (19×19 клеток), Го-моку, Скрабл (Эрудит), Дипломатия, Бридж, Джин Рамми, Шахматы, Шашки, Сёги (японские шахматы), Рэндзю, Китайские шахматы. Участники олимпиады представляли Англию, Германию, Китай, Латвию, Литву, Нидерланды, Россию, США, Тайвань, Финляндию, Францию и Швейцарию.

Из России приехало пять участников. Ветераны компьютерной олимпиады — И. Голубев и А. Мельников — привезли программу «Нейрон», принявшую участие сразу в четырех турнирах: Го (9×9 и 19×19), Го-моку и Рэндзю. А. Лопатин дебютировал с программой «АЛЬФА-бридж», вступив в борьбу с опытнейшими олимпийцами-бриджистами. Программа И. Григорьева «Rummymate» и программа А. Гагина «Джинни» представляли Россию в турнире по Джину Рамми. Интересно, что если на первой, второй и третьей олимпиадах наши соотечест-

венники предпочли поиграть в крестики-нолики в их всевозможных модификациях и провлениях, то в этом году мы сделали мощный рывок в карточных играх.

В итоге, в турнире по бриджу программа А. Лопатина получила третье место, уступив золото германской «Bridge King» и серебро программе «Bridge Вагон», написанной совместно программистами США и Великобритании. В споре за победу по Джину Рамми программа И. Григорьева завоевала первое место, а программа А. Гагина заняла второе.

Турнир по Авари стал захватывающим спором между гигантскими базами данных. Самая большая база данных программы «Marvin» (Швейцария) уступила нидерландской программе «Lithidion», подтвердив ту истину, что ум не всегда определяется только объемом мозга.

Интересные результаты принес шахматный турнир. В нем приняло участие семь программ, в том числе три новичка: нидерландские программы «Genesis» и «Duck» и швейцарская «Ananse». Силы были практически равными, так что в комплекте наград оказались только золотые медали. Их получили три программы: «The King» (Нидерланды), «Hiarcs» (Великобритания) и «Genesis»,

привлекшая к себе особое внимание легкостью побед над шахматными экспертами.

В турнире по Отелло соревновались также семь участников. К сожалению, из немецкой четверки лидеров прошлого года на нынешней олимпиаде была представлена только программа «Aida», оказавшаяся в списке лидеров между двумя французскими программами — «Hotel du Nord» (первое место) и «Jacr'Oth» (третье место).

Большое внимание привлек к себе турнир по Го-моку. Победа программы «Victoria» (Нидерланды) была достигнута с большим трудом. Если бы не неудачи финской «Хокк» (несмотря на постоянные улучшения алгоритма, эта программа из года в год занимает лишь последние места) и не встреча «Victoria» — «Zero Club» (Латвия), в которой латыши умудрились проиграть и за белых и за черных, едва ли можно было однозначно говорить о победе голландского фаворита. Решающая партия с программой Голубева — Мельникова «Нейрон» была выиграна только по времени, да и то потому, что голландцы работали на машине SUN, значительно опережающей ПК по мощностным и скоростным параметрам. Кроме того, в этом году в олимпиаде не участвовали мощные отечественные программы «Matena»

(А. Гитис) и «Stone System» (Н. Александров, А. Григорьев и М. Трухан), ежегодно возглавлявшие турнирные таблицы прошлых олимпиад. Как бы там ни было, «Victoria» заняла первое место, а «Нейрон» и «Polygon» (Нидерланды) поделили второе. Попытка определить более сильную программу из пары серебряных призеров не дала результатов, несмотря на многократные повторные поединки между программами.

В стремительном турнире по Рэндзю, протекавшем буквально перед закрытием олимпиады, золото завоевала программа «Нейрон». Она же взяла и бронзовую медаль по Го (19×19), разделив свой успех с тайваньской программой «Archmage».

В остальном все было как обычно... Упорная борьба за игровыми досками. Напряженные лица программистов, пристально вглядывающихся в экраны мониторов, их шепчущие губы в попытке подсказать правильный ход своим питомцам.

В предпоследний день олимпиады проводились захватывающие матчи между человеком и компьютером. Так, играющая в Дипломат программа француза Даниэля Лёба заняла престижное третье место в матче с шестью экспертами. А нидерландская программа «Shogi», сразившись в двух партиях с Дэвидом Мэрфи, в обеих проиграла. Накануне закрытия олимпиады моя программа сражалась с Дэвидом Леви в Джин Рамми и успешно выиграла у него со счетом 440:93 (что принесло мне даже большее моральное удовлетворение, чем победа на олимпиаде).

В рамках олимпиады проходила IV Лондонская конференция по компьютерным играм, где вы сразу попадали из мира спорта в мир науки. Здесь обсуждались проблемы искусственного интеллекта, излага-

Таблица Комплекты наград, разыгранные на IV Международной компьютерной олимпиаде.

Страна	Золото	Серебро	Бронза	Итого
Нидерланды	5	2	1	8
Великобритания	3	3	1	7
Россия	2	2	2	6
США	1	4	1	6
Тайвань	—	1	2	3
Германия	2	—	—	2
Франция	1	—	1	2
Китай	1	—	—	1
Латвия	—	1	—	1
Швейцария	—	1	—	1
Финляндия	—	—	1	1
Литва	—	—	—	—

лись концепции решения задач программирования логических игр. Спортсмены, в соседнем зале наносившие друг другу сокрушительные удары, на конференциях превращались в мэтров, излагающих основные идеи своих непобедимых алгоритмов, т.е. происходила своего рода закладка фундаментов будущих программ.

Сегодняшнее состояние «думающих» игровых алгоритмов говорит о том, что, возможно, очень скоро игровые программы будут играть лучше людей.

Уже едва не перехватила пальму первенства у чемпиона мира по шашкам Мариона Тинслея программа «Chinook» (см. «Мир ПК», № 10/92). Программа «Victoria», созданная студентами из Лимбургского университета, продемонстрировала решение игры «Connected Four» (Четыре-в-ряд) и трехмерных крестиков-ноликов.

11 августа были объявлены результаты состязания. Что же дальше? Обогатившись новыми знаниями и свежими идеями, участники отправились до-

мой, чтобы продолжить поиск сногсшибательных идей. И как знать, быть может, подходы, найденные при создании компьютерных игр, откроют новые возможности для решения столь актуальных сегодня задач экономического планирования и прогноза.

ОБ АВТОРЕ

Игорь Вячеславович Григорьев — обладатель золотой медали IV Международной компьютерной олимпиады в Лондоне.



**МНОГООТРАСЛЕВАЯ
НАУЧНО-ПРОИЗВОДСТВЕННАЯ
ФИРМА**

МНПФ Rans

официальный дистрибутор фирмы
Intel

в республике Беларусь

*Фирма «Rans» основана в 1992 году
на базе Научно-исследовательского института
электронных вычислительных машин.*

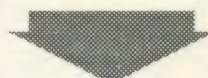
«Rans»:

- ✓предлагает поставку широкой номенклатуры технических и программных средств, выпускаемых фирмой Intel;
- ✓обеспечивает пользователей всей необходимой справочно-методической и технической документацией;
- ✓оказывает консультационные услуги;
- ✓разрабатывает по требованию заказчика документацию: на технические и программные средства ПЭВМ; процессорные платы на базе 80386SX, 80386DX, 80486SX, 80486DX, 80486DX2;
- адаптеры VGA, SVGA;
- адаптеры локальных вычислительных сетей;
- многопортовые адаптеры типа RS-232;
- адаптеры оптоволоконных линий связи;
- встроенные и внешние модемы;
- факсплаты.
- ✓осуществляет установку лицензионных ОС на ПЭВМ;
- ✓проводит исследования, разработку, поставки и сервисное обслуживание компьютерных систем «под ключ» по заказам потребителей.

Оплата за рубли и СКВ.

Наш адрес: г. Минск, ул. Богдановича, 155. Тел.: 34-60-35, 62-51-83. Fax: (0172) 34-94-12

Прикладное ПО на базе СУБД ORACLE
(производство, банки, металлургия, транспорт и др.):
AVALON, ACT, LYRIC и др.



Информационные системы "под ключ"

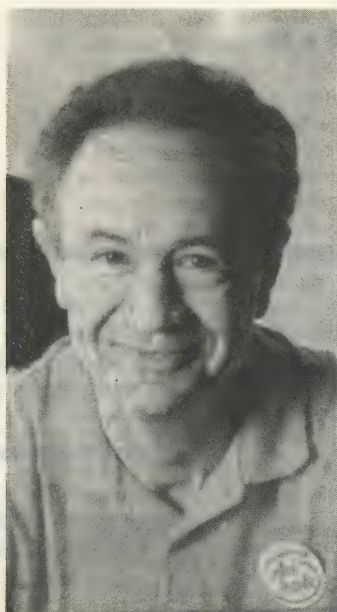
Обучение

Консультирование

Сопровождение

LVS CORPORATION

103045, Москва, Рождественский б. 11/10-9
Тел.: 208-45-31; Факс: 924-09-63



Intel в Москве

И.Б. Рогожкин

Трудно представить себе персональные компьютеры без микропроцессоров Intel.

Еще труднее представить, что легендарный Эндрю Гроув оставит без внимания формирующийся российский рынок ПК.

Активная деятельность корпорации Intel на нашем рынке началась в апреле 1991 г., когда был подписан первый дистрибуторский контракт. Тогда торговое представительство этого мирового гиганта полупроводниковой индустрии насчитывало всего четыре сотрудника и размещалось в одной комнате с одним телефоном. С конца апреля 1992 г. московский офис Intel расположен в здании на ул. Кременчугская, 6/7. Для юридического оформления в соответствии с российским законодательством было создано и зарегистрировано Акционерное общество Intel, для чего были выпущены и проданы на Западе акции Intel Russia. АО Intel — это филиал корпорации Intel, находящийся в полной ее собственности.

Кадры

Деятельностью АО Intel руководит Дмитрий Ротов, региональный менеджер (district manager). Корпорация Intel подразделяет сферу своего международного влияния на семь регионов, одним из которых

является территория бывшего Советского Союза. Выбор руководителя для этого региона не случаен: Дмитрий Ротов обладает огромной энергией, умеет организовать людей и обладает обширными техническими знаниями. Быстрое становление и начало деятельности московского представительства корпорации — во многом его заслуга.

Региональным менеджером по продажам (district sales manager) является Стивен Чейз (Stephen Chase). В штате московского представительства также есть инженеры по продажам (field sales engineers), задача которых — работа с дистрибуторами. Есть специалист по элементной базе и компьютерам, выполняющий роль технического консультанта. Всего в московском офисе сейчас работает 11 человек, причем многие из них набирались по объявлениям в газетах на конкурсной основе. Сначала анкета, потом несколько собеседований, затем испытательный срок. Средний возраст сотрудников около 29 лет. По одному человеку от представительства Intel работают в Санкт-Петербурге, Киеве и Минске и один сотрудник



Новый микропроцессор Intel 486SL

- производительность процессора 486DX*
- пониженное энергопотребление (на 50% ниже, чем 386SL)
- питание 3,3/5 В
- встроенный сопроцессор
- встроенный 32-разрядный контроллер памяти
- средства управления потребляемой мощностью
- малые размеры корпуса

* В настоящий момент имеется 25-МГц версия; появление процессора с тактовой частотой 33 МГц ожидается в январе 1993 г.

помогает представительству в Мюнхене.

Структура

Когда-то в Intel все было не так, как сейчас. Между рядовыми работниками и руководителями высшего ранга было около двадцати промежуточных ступеней. Пять лет назад в корпорации произошла структурная перестройка и теперь осталось всего пять иерархических этажей. Из вертикальной структура превратилась в вертикально-горизонтальную. Система подчинения стала более гибкой, поскольку теперь работа ведется в основном по проектам.

За каждый проект отвечает конкретный человек. Это не

обязательно руководитель высокого ранга, а любой сотрудник, уже имеющий опыт подобной деятельности и соответствующие знания. В каждом проекте участвует множество людей, причем подчинение идет не по вертикали, а по горизонтали. Любой человек может участвовать одновременно в нескольких проектах. Если в одном проекте он руководитель, то в другом руководителем может быть другой человек и подчинение в нем имеет обратное направление. Получается матричная структура, в которой каждый в меру своих способностей участвует в нескольких проектах и в каждом из них играет свою роль. Ни один сотрудник корпорации не чувствует себя всегда

внизу иерархической лестницы.

Одним из наиболее успешно осуществляемых проектов в нетрадиционных областях деятельности корпорации сегодня можно считать программу продвижения на рынок средств расширения ПК. По продажам сетевых адаптеров корпорация Intel вышла на третье место в Европе.

Продуктивная работа такого сложного организма, как корпорация Intel, невозможна без продуманной системы поощрения сотрудников. Она включает как материальные, так и моральные стимулы. Само участие в крупном проекте почетно. Кроме того, внутри фирмы существуют различные звания, например аналогичное нашему «лучший по профессии». Наивысшее звание, которое за всю историю существования корпорации получили только семь человек, — «Intel Fellow». Обладателю этого титула выделяются средства под любой задуманный им проект, причем обоснование проекта не требуется. Поощрение сотрудников, занимающихся продажей изделий, зависит от выполнения намеченного плана. Для технических сотрудников существуют премии за внедрение микросхем Intel в новые изделия. Если какая-либо крупная компания объявляет, что разработала на основе микросхемы Intel совершенно новый продукт и разворачивает его производство, человек, занимавшийся работой с этой фирмой, получает премию.

Внутри корпорации успешно используется соревнование между подразделениями, по итогам которого победителям вручаются призы, а ход соревнования освещается во внутрифирменных изданиях, например в ежеквартальном видеожурнале.

Связь

Для продуктивной работы в новых условиях созданы гибкие и разнообразные средства связи между подразделениями корпорации. Каждый сотрудник Intel может общаться с любым другим сотрудником, находящимся в любой точке земного шара. Для этого есть все необходимое: электронная почта, космическая связь, даже справочник, включающий телефоны всех сотрудников Intel по всему миру.

Обустройство московского офиса корпорации Intel началось именно с организации связи. На Западе подразделения корпорации обычно пользуются электронной почтой cc:Mail, однако в Москве подключиться к ней оказалось невозможно. Нашли замену — систему Sprint-Mail. В результате сейчас корреспонденция посылается в Мюнхен, где через узел Soft-Switch направляется из Sprint-Mail в другие системы электронной почты.

Когда-то в Intel каждый сотрудник пользовался тем программным обеспечением, которое считал для себя наиболее удобным, однако путаница с разными форматами файлов заставила стандартизировать используемое в корпорации ПО. Несмотря на то, что основные сетевые продукты Intel ориентированы на программное обеспечение фирмы Novell, внутри корпорации принято использовать сетевую OS Banyan VINES. В московском офисе работают две сетевые системы: Banyan и Novell. На каждой рабочей станции сети установлен пакет FAXability (разработка Intel). Отдельный компьютер в сети используется как факс-сервер. На факс-сервере работает пакет NET SatisfAXtion, который позволяет посылать или принимать фак-

сы с любой рабочей станции сети. При установке пакета FAXability в рабочей станции устанавливается новое логическое устройство — принтер, например LPT3, которое используется для посылки факсов. Отправлять факсимильные изображения можно из всех программ, выводящих что-либо на принтер, например из текстовых редакторов. При выборе в редакторе команды print появляется окошечко, в котором нужно ввести номер телефона получателя сообщения. Когда номер набран (или вызван из встроенной записной книжки), печатаемый файл передается на факс-сервер, который затем самостоятельно дозванивается и посылает сообщение.

Недавно на здании, в котором размещается московское представительство Intel, появилась двухметровая антенна-тарелка космической связи. На рабочих станциях локальной сети московского офиса стали использоваться новые логические диски. Каждый диск содержит какую-либо информацию, например из отдела маркетинга в Париже. На компьютере он обозначается одной из букв латинского алфавита, как НЖМД или НГМД, а физически располагается на одном из 400 серверов, находящихся в отделениях корпорации во всех уголках мира. Информация о новой продукции, пресс-релизы, файлы с изображениями, которые записываются на диски этих серверов, сразу становятся доступны в Москве. Цель применения космической связи — улучшить качество поддержки дистрибуторов и покупателей, более оперативно отслеживать изменения рынка. Высокая скорость передачи данных по этой линии позволяет передавать не только тексты, но и файлы изобра-

жений. А с помощью слайд-принтера* из файлов изображений оперативно печатаются свежие слайды для презентаций.

Цели и задачи

Если в 1992 г. перед московским представительством корпорации Intel стояли задачи организовать офис и инфраструктуру связи, обучить сотрудников и дистрибуторов и начать продавать продукцию, то в 1993 г. целью является продвижение на рынок новых продуктов и увеличение объема продаж. Среди приоритетных направлений деятельности продаж *микрпроцессоров*, в первую очередь Pentium, и *изделий для расширения персональных компьютеров* (модемы, факс-модемы, математические и инженерные сопроцессоры, процессоры Overdrive, флэш-память, различные контроллеры, карты PCMCIA).

Партнеры

Основными заказчиками и торговыми партнерами АО Intel являются дистрибуторы Intel в странах СНГ. Они обеспечивают рекламу изделий, поиск торговых партнеров, проводят семинары и конференции, работают с конечными пользователями. Продажа изделий не через дистрибуторов, а напрямую через московское представительство Intel возможна только в случае заказа особо крупных партий.

Среди партнеров Intel много различных НИИ, однако обычно они покупают изделия Intel только для целей разработки и маркетинга, т.е. по несколько штук. Крупные партии (иногда даже десятки тысяч изделий) приобретают лишь мощные коммерческие структуры и предпри-

*См. «Мир ПК» № 10/92.

ятия. Наибольшим спросом пользуются процессоры и сопроцессоры. Приобретать комплектующие крупными партиями очень выгодно, поскольку даются большие скидки, однако сегодня из-за спада в промышленности крупных заказчиков мало.

Проявляют интерес к продукции Intel и военные, правда пока только на словах. На продукты, специально разработанные для военных применений, распространяются определенные ограничения. На ряд изделий, например, требуется заявление конечного пользователя о том, для чего он будет их использовать, что они не будут реэкспортироваться и что предназначены для нестратегиче-

ских задач. Для получения подобного документа требуется время.

Коллектив московского представительства корпорации Intel еще очень молод. Учиться работать сотрудникам приходится по ходу дела. Пока они выполняют в основном просветительскую миссию. Устанавливают контакты, рассказывают о своей продукции, показывают образцы товаров. Когда речь заходит о планах на будущее, их зачастую спрашивают, не собирается ли Intel строить в России свои заводы. Пока таких планов нет. И не только потому, что в России нестабильная экономическая ситуация и несовершенное законодательство. В Европе, например, тор-

говое представительство корпорации Intel просуществовало 18 лет до того как было открыто производство. Прежде чем разворачивать производство, необходимо сформировать рынок, способный потреблять его продукцию.

С момента появления в 1972 г. первого микропроцессора пионерские технические решения корпорации Intel привлекают внимание отечественных разработчиков. Теперь же Intel становится не только незримым интеллектуальным партнером, но и партнером деловым.

Контактные телефоны московского офиса Intel:
445-08-23, 444-04-96;
факс: 445-06-96.



АВИАЦИОННАЯ ПРОМЫШЛЕННОСТЬ

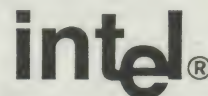
Отраслевой Научно-Исследовательский Центр ПЭВМ "ТЕХНО"

105203, Москва,
ул. Первомайская, 126

Тел.: 485-68-21
Факс: 461-35-14

ОНИЦ ПЭВМ "ТЕХНО" —

дистрибутор фирмы



Наш центр выполняет следующие виды работ:

разработка и поставка специализированных вычислительных систем "под ключ", разработка прикладных многопользовательских систем на основе локальных сетей UNIX, систем электронной почты и базы данных ORACLE, сервис и ремонт ПЭВМ, обучение специалистов и пользователей ПК, информационное и консультационное обслуживание.

The branch research Center "Techno" is a state structure, which is Intel's distributor. "Techno" is dealing with developing and "on-key" delivery of specialized computer systems, developing of applied UNIX-based multiuser's and E-mail systems, ORACLE-based application under UNIX, service and repair of PC's, training of specialists and users of PCs, information & consulting services.



STINS COMAN

STATUS

IN

STATUS



STINS COMAN

Акционерное общество СТИНС КОМАН начало деятельность как дистрибутор компонентов и систем фирмы Intel с марта 1992 года. Наш корреспондент встретился с сотрудником дистрибуторской группы П.Д. Бавевым и задал ему несколько вопросов.

МПК: В чем заключается дистрибуторская деятельность компании СТИНС КОМАН?

Необходимо отметить, что дистрибуция изделий фирмы Intel занимает одно из главных направлений деятельности СТИНС КОМАН. Специалисты, сотрудничающие в настоящее время с фирмой Intel, являются инженерами-разработчиками и имеют опыт работы с продукцией фирмы на предприятиях с различными формами и характером организации производства. Это позволяет сотрудникам группы легко ориентироваться в разных сферах деятельности, где потенциально возможно использование изделий фирмы Intel. При этом работа дистрибуторов строится на следующих принципах:

- продажа систем (рабочие станции, серверы, компьютеры, модемы, средства разработки и т.п.) по образцам;
- предоставление разработчикам доступа к средствам разработки и моделирования;
- оказание практической помощи (прото-типы, техническая консультация, организация производства и сбыта) отечественным производителям средств вычислительной техники, использующих компоненты фирмы;
- для нас одинаково важен как клиент, сделавший заказ на одну микросхему, так и заказавший систему за несколько миллионов;
- предпродажная подготовка по требованиям заказчика;
- гарантийный и послегарантийный сервис;
- адаптация продукции фирмы к особенностям Российского рынка. Так по просьбе фирмы была проведена сертификация модемов Intel в Министерстве связи России.

МПК: Какие продукты фирмы Intel пользуются у Вас наибольшим спросом?

Производителей серийного оборудования и сервисно-ремонтные организации интересуют микросхемы периферии для процессоров i8086, i8088, i80286, ПЗУ с ультрафиолетовым стиранием 2764, 27128, однокристальные микро-ЭВМ серий i486, i51. У разработчиков и малых научно-технических предприятий популярны 16-разрядные микро-ЭВМ i80186, i87196, i87198, риск-процессоры i960, быстродействующая статическая память, телефонные и модемные микросхемы и, конечно же, программируемая логика, программное обеспечение для разработки электронных устройств на их базе.

МПК: Какую продукцию кроме микросхем Вы предлагаете?

Можно выделить три основных вида продукции, с которыми мы работаем. Первый — это сетевое оборудование. Сюда входят адаптеры сети ETHERNET, TOKENRING, специализированный сервер для выполнения архивации (Backup), мощная 32-х разрядная платформа файл-сервера, сервер печати, устройства модемной и факс-модемной связи. Факс-модемная плата SatisFAXtion — сегодня одно из самых популярных изделий. Второй составляют компьютерные платформы на базе процессоров i486. И, наконец, третий — пакеты программного обеспечения. Ни одно устройство не продается без мощной программной поддержки, будь это сетевое оборудование или программируемая логика, или однокристальные микро-ЭВМ. Здесь следует упомянуть об операционной системе iRMX For Windows, которая позволяет на базе обычного персонального компьютера организовать систему, работающую в реальном масштабе времени и использующую мощный графический интерфейс программного обеспечения среды WINDOWS.

МПК: Как осуществляется техническая поддержка поставляемой техники?

Приобретая продукцию фирмы в СТИНС КОМАН, покупатель становится зарегистрированным пользователем, который может воспользоваться всеми имеющимися информационными каналами фирмы. Специалисты СТИНС КОМАН и Intel помогут установить технику, проверить ее работоспособность и проконсультировать при появлении каких-либо проблем.

МПК: Какие цены существуют сейчас на продукцию фирмы Intel и какие предусмотрены формы оплаты?

Продукцию фирмы нельзя отнести к разряду самых дешевых. Например, файл-сервер на базе процессора i486 в приличной конфигурации стоит порядка 15 тыс. долларов. Но за эту цену покупатель получает полную уверенность в том, что эту машину нужно только включить и не беспокоиться о потере данных. Такую же характеристику можно дать другим изделиям фирмы. СТИНС КОМАН продает оборудование на всей территории бывшего Советского Союза и ориентируется на цены, которые рекомендует сам Intel, поэтому они зачастую самые низкие в России. Конечно, цены на какое-либо изделие различаются в зависимости от партии, ритмичности делаемых заказов, возможности заказчика платить авансом, при этом разница в цене может достигать 20-30%.

МПК: Обращаются ли по поводу приобретения комплектующих для ПЭВМ, в частности, на процессоре i286?

Запросы такие есть, но их с каждым месяцем становится все меньше, и это не удивительно — ведь на Западе эти машины в продаже найти достаточно трудно. Что же касается фирмы Intel, то она прекратила производить процессоры i286 и периферийные чипы для них. Более того, наметившееся сокращение производства процессоров i386 не обошло и фирму Intel, хотя она и остается их крупнейшим поставщиком. Пользуясь случаем замечу, что сравнимые по производительности i486SX25 и i386DX33 близки и по стоимости, но компьютер, выполненный на процессоре i486SX25, может быть в любой момент «усилен» до i486DX33 и выше. Для этого необходимо приобрести OverDrive-процессор и установить его в специально предусмотренное для этой цели гнездо, чего нельзя сделать с компьютерами на процессоре i386DX33.

МПК: Наши разработчики всегда страдали от отсутствия справочников и другой технической документации на современную элементную базу. Как Вы решаете эту проблему?

У нас можно получить информацию о всех материалах, которые издаются фирмой, и заказать их оригиналы или приобрести копии. Предоставляется возможность ознакомиться с имеющейся в нашей библиотеке документацией.

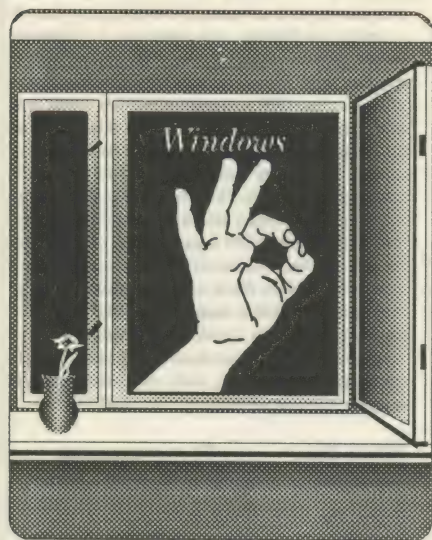
МПК: Насколько нам известно, средства разработки, поставляемые фирмой Intel, имеют достаточно высокую цену и разработчику трудно решиться заплатить такую цену для того, чтобы спроектировать одно-два устройства. Как можно решить эту проблему?

Специально для этих целей в СТИНС КОМАН организовано несколько рабочих мест для разработчиков, в частности, для проектирования электронных схем на микросхемах программируемой логики, для разработки программных средств в среде iRMX for Windows. Планируется оснащение рабочего места для отработки контроллеров на однокристальных микро-ЭВМ. Все эти рабочие места оснащены необходимой аппаратурой и программным обеспечением.

МПК: Не могли бы Вы оценить результаты деятельности Вашей фирмы как дистрибутора Intel?

Основной показатель для нас, конечно, объем продаж, который достиг к концу 1992 г. 350 млн. руб. Но кроме этого показателя существует еще динамика роста объема продаж и количество постоянных заказчиков. Особую гордость вызывает то обстоятельство, что мы их не теряем.

Телефон: 465-64-08, 465-69-22
Факс: 465-90-34



Таинственный редактор PIF-файлов

Брайан Ливингстон

Мы продолжаем публикацию журнального варианта книги Б. Ливингстона «Windows 3.1 Secrets».

Одна из служебных программ Windows — редактор PIF Editor — позволяет создавать файлы информации о программе (Program Information Files — PIF), используемые для запуска прикладных программ DOS из среды Windows.

Зная возможности этой программы, вы сможете повысить производительность сеанса DOS.

Пожалуй, ни одна из входящих в комплект поставки Windows служебных программ не вызывает такого чувства неуверенности, как этот маленький редактор. И немудрено: вам предлагается задать множество параметров, о назначении многих из которых неискушенный пользователь предпочитает не знать, а оценить результаты своего труда оказывается задачей едва ли не более сложной.

НАЧНЕМ СНАЧАЛА

Итак, что же скрывается за неприметной, на первый взгляд, пиктограммой? Редактор PIF-файлов (PIF Editor) — средство, позволяющее задавать параметры сеанса DOS под управлением Windows для различных «старых приложений» (old apps — так в документации к системе именуются программы DOS). Самыми важными являются, естественно, параметры, управляющие приоритетом задачи и выделением всех типов памяти. Другие параметры позволяют повысить производительность сеанса DOS или системы в целом и сэкономить немного памяти для иных нужд. А если программа использует в своих целях зарезервированные системой Windows команды клавиатуры, например <Ctrl> + <Esc> или <Alt> + <Tab>, с помощью редактора PIF-файлов можно вернуть им в программе «правильное» поведение.

Учтите, что все изложенное относится только к программам для DOS; программы для Windows сами распоряжаются оперативной памятью и процессором, и вы можете задать лишь приоритет виртуальной машины Windows в целом, но не каждой программы в отдельности.

Редактор PIF-файлов, как следует из его названия, хранит относящуюся к каждой из программ информацию в отдельном файле с расширением PIF. Эти файлы, подобно исполняемым (.EXE и .COM) и командным .BAT-файлам, могут быть использованы для запуска описанных в них программ из оболочки Windows — командой Run из меню File или с помощью пиктограммы. Такая возможность появляется даже в сеансе DOS под управлением Windows: если в командном файле указать имя PIF-файла, соответствующая ему программа будет запущена с учетом всех заданных параметров.

КАЖДОЙ ВЕЩИ — СВОЕ МЕСТО

Приступая к созданию PIF-файлов, постарайтесь не устраивать беспорядка в каталоге WINDOWS. Конечно, оптимальным является вариант, при котором этот каталог не содержит никаких файлов, кроме тех, что появились при установке системы. Важное правило при работе не только с Windows, но и с любой другой программой — не допускать хранения в одном и том же каталоге файлов данных и исполняемых файлов, обычно предназначенных только для чтения. Соблюдение этого правила поможет вам решить многие проблемы и, в частности, защититься от вирусов.

Итак, создайте новый каталог C:\PIF и позаботьтесь о том, чтобы он был указан в качестве параметра команды PATH раньше, чем каталог C:\WINDOWS. Затем укажите C:\PIF в качестве рабочего каталога программы PIFEDIT.EXE. В каталоге C:\WINDOWS уже есть файлы с расширением PIF — не забудьте и их перенести на новое место.

ПОРА ЗА ДЕЛО...

На самом деле вы с первого дня работы с Windows пользуетесь PIF-файлами, даже не подозревая об их существовании. А между тем, содержимое файла _DEFAULT.PIF, используемого по умолчанию при запуске любой программы DOS, заслуживает максимум «двойку с плюсом». Разработчики фирмы Microsoft по своему обыкновению перепутали среднее с худшим, создав этот файл в расчете на некую особо «злостную» программу DOS, работающую с дисплеем EGA в графическом режиме и требующую одновременно все типы памяти — отображаемую и расширенную.

Разумеется, показатели быстродействия оказываются здесь близкими к худшим из возможных. Ниже будет показано, как можно задавать параметры в расчете на более реальную ситуацию. Но и в этом случае установки самого редактора PIF-файлов по умолчанию будут полностью соответствовать первоначальному содержанию _DEFAULT.PIF, т.е. наихудшему варианту, что уже само по себе является достаточным основанием для того, чтобы очень внимательно изучить возможности программы.

Возможности же ее гораздо шире, чем может показаться на первый взгляд. Любой PIF-файл по существу содержит два набора параметров, соответствующие разным режимам работы Windows. На рисунках показаны диалоговые окна редактора для стандартного (рис.1а) и расширенного (рис.1б) режимов, а также окно Advanced Options расширенного режима (рис. 2). Многие параметры присутствуют в диалоговом окне редактора и в том и в другом случаях, но это вовсе не означает, что, например, резервирование клавиш <Alt> + <Tab>, установленное в стандартном режиме, будет действительно и в расширенном. В табл. 1 перечислены все общие параметры с указанием того, допускают ли они различные значения в стандартном и расширенном режимах.

В данной статье рассматриваются назначение и использование многочисленных параметров, которые могут быть установлены с помощью редактора PIF-файлов. Вначале описываются параметры, присутствующие как в стандартном, так и в расширенном режимах, а затем — параметры, являющиеся уникальными для каждого из режимов работы системы. Во всех случаях даются рекомендации по установке значений для различных типов приложений DOS. Имеется и общая справочная таблица (см. табл. 2), которую полезно иметь под рукой при создании PIF-файлов. Отдельный раздел посвящен программам, требующим к себе особого отношения, — таковы последние версии некоторых популярных пакетов, способные самостоятельно взаимодействовать с Windows.

ОБЩИЕ ПАРАМЕТРЫ

Установки, описываемые в данном разделе, являются общими для стандартного и расширенного режимов.

Program Filename (имя файла программы)

На первый взгляд, параметр Program Filename представляется весьма простым, но и он имеет свои «секреты». Информацию о настройке программы можно сохранить в PIF-файле с произвольным именем, отличающимся от имени ее исполняемого файла. Между тем, одинаковые имена дают определенные преимущества. Если программа запускается из Администратора или Администратора файлов (например, из меню File Run), Windows сначала ищет файл с тем же именем и расширением PIF и в случае успеха автоматически использует указанные в нем параметры. Если же такой файл не найден, система обратится к _DEFAULT.PIF — его содержимое читателю уже известно.

В любом варианте должно быть задано имя реально

Таблица 1. Общие параметры стандартного и расширенного режимов.

Параметр	Возможны ли различные значения в стандартном и расширенном режимах
Program Filename (имя файла программы)	Нет
Window Title (заголовок окна)	Нет
Optional Parameters (параметры командной строки)	Да
Start-up Directory (рабочий каталог)	Нет
Video Mode (режим дисплея)	Да
Memory Requirements (требования к памяти)	Да
XMS (Extended) Memory (расширенная память)	Да
Close Window on Exit (закрывать окно при завершении)	Нет
Reserve Shortcut Keys (резервировать «горячие клавиши»)	Да

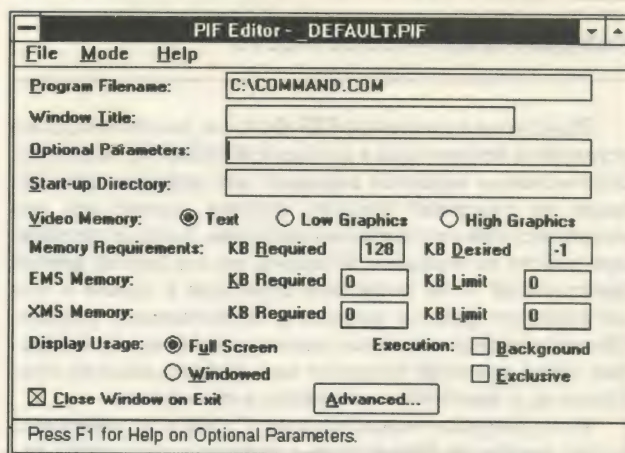
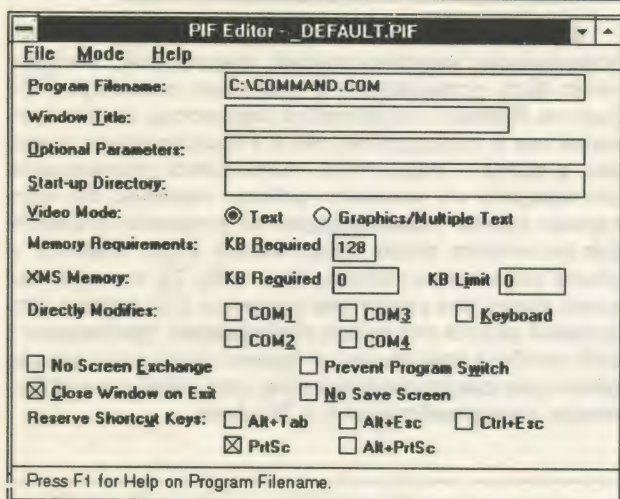


Рис. 1. Окно редактора PIF-файлов при задании параметров: а) для стандартного режима; б) для расширенного режима.

существующего на диске исполняемого файла — даже при редактировании _DEFAULT.PIF, для которого содержимое этого поля не имеет значения. Программа не позволит сохранить редактируемый PIF-файл до тех пор, пока не будет указано имя существующего файла.

Window Title (заголовок окна)

Строка текста, заданная в поле Window Title, появится в панели заголовка окна программы или станет подписью под пиктограммой, если окно минимизировано. Параметр можно не указывать — в таком случае Windows использует имя PIF-файла (без расширения PIF).

Optional Parameters (параметры командной строки)

Здесь задаются параметры, которые передаются программе при запуске. Однако если напечатать что-либо в строке меню File Run после имени PIF-файла, содержимое поля Optional Parameters не будет принято в расчет.

Не все знают, что в строке Optional Parameters можно указать вопросительный знак — в этом случае при запуске программы появится диалоговое окно, содержащее запрос на нужные параметры. К сожалению, эта полезная возможность в Windows 3.0 реализована небезупречно — передать параметры можно .COM- и .EXE-, но не .BAT-файлу.

Start-Up Directory (рабочий каталог)

В технической документации к Windows 3.0 настоятельно рекомендуется использовать поле Start-Up Directory для задания каталога, который необходимо сделать текущим перед запуском программы. Однако существует одна незадокументированная особенность — если создать ассоциацию какого-либо расширения с PIF-файлом, скажем, с помощью строки DOC=WORD.PIF ^.DOC

в разделе [Extensions] файла WIN.INI (для загрузки любого DOC-файла с одновременным запуском Word for DOS при двойном щелчке «мыши» на имени DOC-файла в File Manager), то при задании любого каталога в качестве параметра Start-Up Directory нужный документ загружен не будет. Хотя в версии 3.1 данная ошибка устранена, наиболее надежным способом запуска программы из указанного каталога представляется создание .BAT-файла, изменяющего текущий каталог до загрузки соответствующей программы.

Video Mode (режим дисплея)

Параметр Video Mode определяет использование графического или текстового режимов дисплея при работе DOS-программы. В стандартном режиме пользователь может установить значение Text (текстовый режим) или Graphics/Multiple Text (графика/многостраничный текст). В расширенном режиме предлагаются три варианта: Text, Low Graphics (графика низкого разрешения) или High Graphics (графика высокого разрешения). Однако даже для графической программы можно установить значение Text — это не только не помешает ее выполнению, но и увеличит объем доступной программе основной памяти. Проблемы могут появиться лишь при попытке переключиться на другую программу: вполне вероятно, что придется сначала завершить выполнение текущей. Но если вы все-таки попадете в Windows и запустите еще один сеанс DOS, система может «отобрать» видеопамять у первой программы, сделав возврат в нее невозможным. Поэтому значение Video Mode: Text наиболее подходит для программ, использующих графический режим, хотя вы и не сможете переключаться из них на другие приложения.

Значение Graphics/Multiple Text подойдет для программ, работающих в многостраничном текстовом режиме. Этот режим требует 32 Кбайт видеопамати, где помещаются до восьми страниц текста (25 строк × 80 колонок занимают 2000 байт, плюс столько же для задания атри-

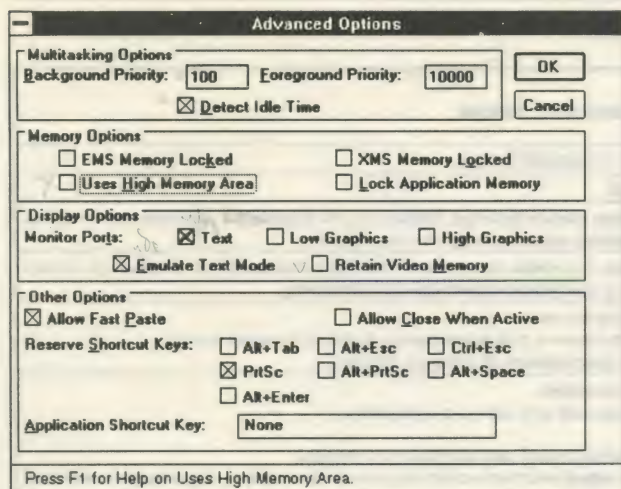


Рис. 2. Окно Advanced Options позволяет установить еще несколько параметров DOS-программы в расширенном режиме.

бута каждого символа; итого — 4000 байт на экранную страницу), которые программа подготавливает заранее и затем может мгновенно вывести на экран. Многие текстовые редакторы используют данную возможность для быстрого перехода к различным участкам текста. Такого же объема видеопамати требуют и программы, применяющие графику разрешения CGA.

В расширенном режиме имеется дополнительный переключатель Retain Video Memory (удерживать видеопамать), также влияющий на использование видеопамати. Если он не установлен, то при переходе в Windows и запуске еще одной программы системе может не хватить памяти для возврата к первой программе. Напротив, если этот переключатель установить, то памяти может не хватить уже для запуска второй программы.

Memory Requirements (требования к памяти)

Как известно, в стандартном режиме можно запустить не более одной программы DOS, и на время ее работы выполнение всех приложений Windows будет приостановлено. Зато она получит при запуске весь свободный объем основной памяти. Поэтому значение, превышающее 128 в поле KB Required, не приведет к увеличению доступного программе объема основной памяти.

Иначе обстоит дело с расширенным режимом. Если программа требует для своего выполнения определенного объема памяти и не способна использовать больший ее объем с целью повышения производительности, есть все основания указать в строке KB Required минимальную величину — 128 Кбайт. В таком случае программа будет загружаться быстрее, а неиспользуемая память станет доступна другим приложениям. Если же установить значение равным -1, система заставит все Windows-программы освободить максимально возможный объем основной памяти и передаст его

DOS-программе. Это значение можно указать и в поле KB Desired, появляющемся только в расширенном режиме Windows.

XMS Memory (расширенная память)

Расширенная память используется самой Windows, и обращение к ней DOS-программы в обход стандартных функций системы может привести к нарушениям работы последней. Но некоторые DOS-программы способны задействовать расширенную память, не мешая Windows работать. При этом они поддерживают стандарт DPMI (DOS Protected Mode Interface — интерфейс DOS в защищенном режиме): разработанный фирмой Microsoft стандартный способ работы с расширенной памятью в многозадачной среде. В Windows доступ к расширенной памяти обеспечивает драйвер HIMEM.SYS, но могут применяться и более совершенные программы — диспетчеры расширенной памяти, например QEMM386.SYS фирмы Quarterdeck.

Если же программа не способна использовать расширенную память описанным образом, следует указать нулевое значение. Программа в таком случае дополнительно получит в свое распоряжение несколько килобайт основной памяти.

В стандартном режиме, в отличие от расширенного, нет возможности выделить программе определенный объем отображаемой памяти. Вся имеющаяся в системе отображаемая память автоматически выделяется программе DOS при ее запуске.

ПАРАМЕТРЫ СТАНДАРТНОГО РЕЖИМА

Описываемые ниже параметры управляют выполнением DOS-программ в реальном и стандартном режимах.

Directly Modifies (непосредственно изменяет)

Если программа не использует стандартные функции DOS и BIOS, а работает непосредственно с клавиатурой либо последовательными портами, подобно многим игровым и коммуникационным программам, следует установить один или несколько переключателей группы Directly Modifies. В такой ситуации Windows не позволит запустить программу, если это может привести к конфликту между различными приложениями и краху системы. Кроме того, установка любого из переключателей будет препятствовать переключению программ — ведь у Windows нет способа «узнать» о том, что программа, возможно, изменила параметры последовательного порта или заполнила чем-то буфер клавиатуры.

No Screen Exchange (запретить обмен с экраном)

При установке режима No Screen Exchange вы лишаетесь возможности копировать содержимое экрана в системный буфер clipboard с помощью клавиш <PrintScreen> и <Alt> + <PrintScreen>, зато программа получает немного свободной памяти. Того же

Таблица 2. Справочная таблица параметров PIF-файлов.

Стандартный (и реальный) режим	
Program Filename	Полное имя исполняемого файла, например C:\COMMAND.COM.
Window Title	Заголовок окна или подпись под пиктограммой.
Optional Parameters	Любые параметры командной строки, например имена файлов. Указание «?» позволяет интерактивно задать параметры при запуске. Для расширенного режима возможны другие значения.
Start-Up Directory	Указанный каталог становится текущим перед запуском программы. Использование этого параметра может помешать загрузке файла по его расширению в требуемую прикладную программу.
Video Mode: Text Graphics/Multiple Text	Для расширенного режима возможны другие значения. При запуске программы резервируется область памяти для размещения одной страницы текста. Резервируются 32 Кбайт для графики низкого разрешения (8 страниц текста).
Memory Requirements: KB Required XMS Memory KB Required KB Limit	Для расширенного режима возможны другие значения. Минимальный объем основной памяти, необходимый для запуска программы. Расширенная память. Минимальный объем расширенной памяти, необходимый для запуска программы. Максимальный объем расширенной памяти, который система может выделить программе; -1 для выделения всей расширенной памяти.
Directly Modifies: COM Ports Keyboard	Необходимо использовать при запуске программ, работающих с портами и клавиатурой в обход стандартных функций системы и BIOS. Установка одного из переключателей делает невозможным переход к другой программе. При установке переключателя Windows работать с указанным портом будет только DOS-программа. При установке переключателя Windows не сохраняет текущие режимы и содержимое экрана.
No Screen Exchange	Установка переключателя запрещает копирование содержимого экрана в буфер и позволяет сэкономить немного памяти.
Prevent Program Switch	Делает невозможным переход к другой программе.
Close Window on Exit	Если переключатель установлен, то по завершении программы происходит возврат в Windows; в противном случае — выход в командный процессор DOS.
No Save Screen	Если переключатель не установлен, Windows сохраняет и восстанавливает содержимое экрана DOS-программы при переключении между задачами.
Reserve Shortcut Keys	Предоставляет программе в исключительное пользование следующие комбинации клавиш: <Alt> + <Esc>, <Alt> + <PrtSc>, <Alt> + <Tab>, <Ctrl> + <Esc>, <PrtSc>. В расширенном режиме возможны другие установки этих клавиш.
Расширенный режим — основные параметры	
Program Filename	Полное имя исполняемого файла, например C:\COMMAND.COM.
Window Title	Заголовок окна или подпись под пиктограммой.
Optional Parameters	Любые параметры командной строки, например имена файлов. Указание «?» позволяет интерактивно задать параметры при запуске. Для стандартного режима возможны другие значения.
Start-Up Directory	Указанный каталог становится текущим перед запуском программы. Использование этого параметра может помешать загрузке файла по его расширению в требуемую прикладную программу.
Video Memory: Text Low Graphics High Graphics	Для стандартного режима возможны другие значения. При запуске программы резервируется область памяти размером около 16 Кбайт. Резервируется 32 Кбайт для графики низкого разрешения (CGA). Резервируется 128 Кбайт памяти для работы с адаптерами EGA и VGA.
Memory Requirements: KB Required KB Desired	Для стандартного режима возможны другие значения. Минимальный объем основной памяти, необходимый для запуска программы; -1 для выделения всего доступного объема. Максимальный объем основной памяти, который система может выделить программе; -1 для выделения всего доступного объема.

результата можно достичь с помощью переключателей Reserve Shortcut Keys, описанных ниже. Следует лишь помнить о том, что стандартный режим не позволяет в принципе копировать графический экран DOS в буфер — переносить можно только текст.

Prevent Program Switch (запретить переключение программ)

Если вы не собираетесь переходить к другим задачам после запуска DOS-программы, есть все основания установить режим Prevent Program Switch. В этом

случае программе будет выделен дополнительный объем основной памяти.

No Save Screen (не сохранять содержимое экрана)

В стандартном режиме система всегда сохраняет содержимое экрана, если пользователь переключается из сеанса DOS к Windows-программе, и восстанавливает экран при обратном переключении. DOS-программа, как правило, «не подозревает» об этих операциях, поэтому, если не сохранять содержимого экрана, пользователь при возврате в сеанс DOS увидит чистый дисплей.

Продолжение табл. 2.

EMS Memory: KB Required KB Limit	Отображаемая память. Минимальный объем отображаемой памяти, необходимый для запуска программы. Минимальный объем отображаемой памяти, который система может выделить программе; -1 для выделения всей отображаемой памяти.
Locked	Установка переключателя запрещает системе сбрасывать на диск содержимое принадлежащих программе областей EMS, что ускоряет переключение, но может сделать невозможным запуск еще одной программы.
XMS Memory: KB Required KB Limit	Расширенная память; для стандартного режима возможны другие значения. Минимальный объем расширенной памяти, необходимый для запуска программы. Максимальный объем расширенной памяти, который система может выделить программе; -1 для выделения всей расширенной памяти.
Locked	Установка переключателя запрещает системе сбрасывать на диск содержимое принадлежащих программе областей XMS, что ускоряет переключение, но может сделать невозможным запуск еще одной программы.
Display usage: Full Screen Windowed	Программа запускается на полном экране. Программа запускается в окне; переключение между двумя режимами производится с помощью <Alt> + <Enter>.
Execution: Background Exclusive	Позволяет программе работать в фоновом режиме с приоритетом Background Priority. При установленном переключателе программа получает наивысший приоритет, приостанавливая выполнение других задач: если она выполняется на полном экране, времени не получает даже Windows.
Close Window on Exit	Если переключатель установлен, то по завершении программы происходит возврат в Windows; в противном случае — выход в командный процессор DOS.
Расширенный режим — окно Advanced Options	
Multitasking Options: Background Priority Foreground Priority Detect Idle Time	Устанавливает относительную величину кванта времени, получаемого программой в активном и фоновом режимах; допустимые значения 0—10000. Значение для фонового режима; не учитывается, если переключатель Execution:Background не установлен. Значение для активного режима. При установке переключателя Windows перестает выделять время бездействующей программе.
Uses High Memory Area	Позволяет программам, способным работать с высокой памятью, использовать ее совместно.
Lock Application Memory	Установка переключателя запрещает системе сбрасывать на диск содержимое участка основной памяти, принадлежащего программе, что ускоряет переключение, но может сделать невозможным запуск еще одной программы.
Monitor Ports: Text Low Graphics High Graphics	Необходимо использовать, если Windows не может правильно восстановить изображение при переключении программ, работающих с дисплеем необычным образом. Используется для программ, работающих в текстовом режиме. Необходимо программам, использующим графику CGA. Устанавливается, если программа работает с адаптером EGA.
Emulate Text Mode	Увеличивает скорость вывода на экран в текстовом режиме, но у некоторых программ может привести к нарушению изображения.
Retain Video Memory	Установка переключателя резервирует не используемую программой видеопамять, которая могла бы быть «захвачена» другой программой, что не позволило бы перейти в графический режим.
Allow Fast Paste	Не следует устанавливать, если программа не способна обработать вводимые символы с той скоростью, с которой их посылает системный буфер.
Allow Close When Active	Позволяет завершать программу без участия ее собственной команды завершения; использовать только для программ, стандартным образом работающих с файловой системой DOS (с использованием обработчиков).
Reserve Shortcut Keys	Предоставляет программе в исключительное пользование следующие комбинации клавиш: <Alt> + <Esc>, <Alt> + <PrtSc>, <Alt> + <Tab>, <Ctrl> + <Esc>, <PrtSc>, <Alt> + <Enter>, <Alt> + пробел. В стандартном режиме возможна установка других переключателей.
Application Shortcut Key	Нажатие указанной комбинации клавиш позволяет сделать уже загруженную программу активной.

Некоторые программы, однако, способны определить наличие Windows в памяти компьютера и обработать сообщения, посылаемые при переключении. В этом случае содержимое экрана будет сохранено и затем восстановлено без участия Windows, а установка параметра No Save Screen сэкономит драгоценную память для других нужд.

Reserve Shortcut Keys (резервировать «горячие клавиши»)

Догадаться о назначении переключателей Reserve Shortcut Keys ни для кого не составит труда. Если, например, DOS-программа использует команду клави-

атуры <Alt> + <Tab> для собственных целей, Windows должна быть поставлена об этом в известность, иначе нажатие указанных клавиш приведет, как обычно, к переключению на предыдущую активную задачу. Если же все переключатели в данном разделе будут установлены, у пользователя не останется способов переключения задач: эффект будет таким же, как и при установке переключателя Prevent Program Switch.

Нужно отметить, что в Windows 3.0 при работе в стандартном (и реальном) режиме попытка использовать клавишу <PrintScreen> в сеансе DOS для вывода содержимого экрана на принтер обречена на неудачу. Следовательно, устанавливать переключатели для этой клавиши имеет смысл лишь в версии 3.1.

ПАРАМЕТРЫ РАСШИРЕННОГО РЕЖИМА

Расширенный режим обеспечивает значительно большие возможности управления сеансом DOS.

Display Usage (использование дисплея)

В расширенном режиме пользователь может выбирать между выводом на полный экран (Full Screen) или в окно (Windowed). Второй режим предъявляет несколько завышенные требования к памяти, и это может стать причиной неудачи при запуске программы в Windows 3.1. Версия 3.0 не поддерживает графический режим программ DOS в окне, поэтому следует убедиться в том, что программа не переходит в данный режим даже на короткое время.

Execution (режим выполнения)

Устанавливая переключатели Background (фоновый режим) и Exclusive (режим исключительного использования ресурсов компьютера), пользователь имеет возможность управлять не только выполнением программы, но и работой системы в целом.

Если переключатель Background не установлен, программа вообще не будет выполняться до тех пор, пока не станет активной. Такой режим предпочтителен для всех программ, которые не совершают каких-либо полезных действий в фоновом режиме. Возможны и другие случаи: по сведениям разработчиков фирмы Lotus, пакет Lotus 1-2-3 версии 3.1 в этом режиме может неправильно работать. Но даже способная производить фоновые вычисления программа может не получить доступа к процессору, если выполняется одно из следующих условий:

- активна еще одна программа DOS, для которой установлен режим Exclusive;
- другая программа DOS имеет приоритет 10000 в активном режиме и производит какие-либо вычисления;
- в разделе [386Enh] файла SYSTEM.INI имеется строка WINEXCLUSIVE=TRUE, указывающая на то, что при работе с Windows-программой (в расширенном режиме) выполнение всех DOS-программ подавляется.

Установка переключателя Exclusive приостанавливает выполнение не только всех остальных DOS-программ, но и Windows-программ. Это приводит к нарушению работы таймеров, коммуникационных программ и других приложений, которым необходим постоянный доступ к ресурсам компьютера. Впрочем, если DOS-программа выполняется в окне, а не на полном экране, Windows-программы все-таки получат требуемое им время. Самый лучший способ добиться максимальной производительности DOS-программы без ущерба для прочих задач — выключить режим Exclusive и установить приоритет в активном режиме (Foreground Priority) равным 10000.

Advanced Options (дополнительные параметры)

Нажав экранную кнопку Advanced..., вы попадаете в

новое окно редактора PIF-файлов Advanced Options, позволяющее задать еще несколько важных параметров.

Multitasking Options (параметры многозадачности)

Не вдаваясь в подробное обсуждение понятия многозадачности, приведем общие рекомендации для установки значений Background Priority (приоритет программы в фоновом режиме) и Foreground Priority (то же в активном режиме).

К сожалению, программ, которые определяли бы наиболее эффективные значения приоритетов для сеанса DOS в зависимости от вычислительной мощности компьютера, пока не существует. Казалось бы, максимальный приоритет должен соответствовать наивысшей производительности, однако после некоторого значения приоритета производительность практически перестает расти. Эту цифру можно определить с помощью BAT-файла, состоящего, например, из многократно повторенной команды DIR. Засекая время его выполнения при различных значениях приоритета в активном и фоновом режимах, можно довольно точно установить оптимальные значения. Если же у вас нет тяги к экспериментам, используйте для активного режима максимально возможное значение — 10000, а для фонового — 100. Результаты же тестов показывают, что гораздо заметнее влияет на производительность значение Minimum Timeslice — системного кванта времени*.

Detect Idle Time (контроль бездействия)

Довольно часто программа, получив управление, расходует свой квант времени на ожидание ввода с клавиатуры. Windows может попытаться определить эту ситуацию и перестать выделять время бездействующей программе до тех пор, пока ей не будут адресованы какие-либо команды. Удастся это системе или нет — зависит от самой программы.

Есть, впрочем, стандартный способ сообщить системе о том, что программа не занята: занести в регистр AX шестнадцатеричное число 1680h и вызвать прерывание 2F. Большинство DOS-программ, конечно, этого не делают. Для того чтобы не попасть в затруднительное положение, нужно руководствоваться следующими правилами.

1. Если программа в фоновом режиме не выполняет никакой существенной работы, переключатель Detect Idle Time должен быть установлен.

2. Если программа делает что-то через определенные промежутки времени или по наступлении события, переключатель устанавливать нецелесообразно.

3. Если программа сообщает Windows о своей занятости описанным выше стандартным способом, переключатель также необходимо сбросить.

EMS Memory (отображаемая память)

Два поля, KB Required (требуемый объем памяти) и KB Limit (предельный объем памяти), позволяют зада-

* Эта тема будет подробнее рассмотрена в одной из следующих статей цикла.

вать соответственно минимальный и максимальный объемы отображаемой памяти, которая выделяется программе. Их значения, впрочем, не вполне соответствуют наименованию полей.

Предположим, ваша программа не требует обязательного наличия отображаемой памяти, но способна ее использовать. На первый взгляд, разумно было бы установить значение 0 для KB Required и 1024 для KB Limit. При этом, однако, программа никогда не получит отображаемой памяти, поскольку Windows будет ожидать особого запроса на ее выделение. Большинство же программ сначала пытаются самостоятельно определить объем доступной им памяти, и он оказывается равным нулю, так как память еще не выделена. После этого, естественно, запрос на выделение памяти не поступит. В такой порочный круг попала, например, программа Lotus 1-2-3 версии 2.01.

Lock Memory (блокировка памяти)

Три независимых переключателя позволяют блокировать основную, отображаемую и расширенную память, выделяемую программе — соответственно Lock Application Memory, EMS Memory Locked и XMS Memory Locked. Если участок памяти заблокирован, система не сможет сбросить его содержимое на диск после того, как вы переключились на другую задачу. Разумеется, обратное переключение при этом произойдет быстрее, но вам, возможно, не удастся запустить еще одну программу из-за нехватки памяти. Чтобы не осложнять работу диспетчера памяти, блокировать память следует только в исключительных случаях, когда программа непрерывно работает с памятью, и сброс последней на диск может привести к нежелательным последствиям.

Uses High Memory Area (использует область высокой памяти)

Высокая память — первые 64 Кбайт расширенной памяти — единственная ее область, к которой имеют доступ DOS-программы в реальном режиме, но, к сожалению, такая возможность используется ими довольно редко. Драйвер HIMEM.SYS, управляющий памятью в Windows, позволяет занять эту область любой программой, посылающей соответствующий запрос.

Если область высокой памяти свободна, то под управлением Windows ее смогут использовать несколько программ, для чего переключатель Uses High Memory Area должен быть установлен. При сброшенном переключателе программа, в принципе способная работать с высокой памятью, не сможет ею воспользоваться.

Правда, высокая память может быть «захвачена» DOS-программой еще до запуска Windows и становится в этом случае недоступной другим программам.

Monitor Ports (отслеживание состояния портов)

Переключатели Monitor Ports управляют отнюдь не последовательными и параллельными портами вашего компьютера, а режимом работы видеоадаптера, который DOS-программа может переключать в другой ре-

жим непосредственно, не прибегая к стандартным функциям операционной системы. Не зная об этих изменениях, Windows сохраняет и восстанавливает содержимое экрана в соответствии со своими «представлениями» о текущем режиме видеоадаптера. Наибольшие проблемы это вызывает при использовании EGA-дисплея.

Если какой-либо переключатель Monitor Ports установлен, система активизирует небольшую резидентную программу, которая отслеживает все обращения к видеоадаптеру, существенно замедляя вывод на экран. К сожалению, именно этот режим установлен как в файле DEFAULT.PIF, так и при создании нового PIF-файла. Между тем, переключатели Text и Low Graphics практически бесполезны, поскольку при работе в текстовом режиме и с графикой низкого разрешения проблемы обычно не возникают. Переключатель High Graphics может пригодиться лишь в том случае, если программа использует EGA-графику и при восстановлении содержимого экрана возникают ошибки.

Emulate Text Mode (эмуляция текстового режима)

При установке переключателя Emulate Text Mode система Windows заменяет функции BIOS ПЗУ, работающие с экраном в текстовом режиме, своими собственными, более быстрыми процедурами. Если же при этом нарушается изображение или перестает правильно работать «мышь», переключатель лучше вернуть в начальное состояние.

Retain Video Memory (удерживание видеопамати)

О назначении переключателя Retain Video Memory речь шла выше, при описании параметра Video Mode.

Allow Fast Paste (режим быстрой вставки)

Расширенный режим позволяет передать DOS-программе содержащийся в системном буфере текст. В режиме быстрой вставки символы посылаются в программу с максимальной возможной скоростью, и она может не справиться с их обработкой. В этой ситуации переключатель Allow Fast Paste устанавливать не следует.

Allow Close When Active (возможность принудительного завершения)

Включение режима Allow Close When Active позволяет завершить DOS-программу, не пользуясь ее собственной командой выхода и даже не отвлекаясь от работы в Windows. Для этого необходимо, щелкнув «мышью» на нужной пиктограмме, выбрать в появившемся меню пункт Close. Если программа выполняется в окне, того же результата можно добиться двойным щелчком в панели системного меню. В обоих случаях появится диалоговое окно с просьбой подтвердить данное решение. Гораздо больше времени потребовалось бы на выход из сеанса DOS с помощью команды Exit — единственный способ, если режим Allow Close When Active не задействован.

«Быстрый» выход из программы, однако, таит в себе

большую опасность. Если в момент завершения программа имеет открытые файлы, результат не всегда можно предсказать. Большинство программ при работе с файлами использует обработчики (handles), и в этом случае Windows сможет проследить за тем, чтобы данные на диске не пострадали. Но если программа задействует устаревшие FCB (File Control Blocks) или вообще не использует функции DOS, данные могут быть безвозвратно потеряны. Последнее относится к ранним версиям многих популярных пакетов, например DataEase 2.1 и WordStar 3.3. Лучше вообще избегать «быстрого» выхода из программ, работающих с файловой системой, — как говорится, береженого Бог бережет.

Reserve Shortcut Keys (резервирование «горячих клавиш»)

В дополнение к одноименной группе переключателей стандартного режима можно зарезервировать команды <Alt> + пробел и <Alt> + <Enter> (не используемые Windows в стандартном режиме). Досадным упущением является то, что Windows не оставляет возможности сохранить за программой использование клавиши <Alt>, нажатие которой в системе активизирует меню.

Application Shortcut Key («горячая клавиша» программы)

Нажатие указанной в качестве параметра Application Shortcut Key комбинации клавиш вызовет мгновенное переключение на данную программу (разумеется, если она уже была запущена). К сожалению, многие пакеты, например Word for Windows, используют почти все сочетания буквенно-цифровых клавиш с <Ctrl> и <Alt>. Самым безопасным оказывается сочетание любой клавиши с <Ctrl> + <Shift>.

Если вас угораздило-таки переопределить уже «занятое» другой программой сочетание клавиш, избавиться от неудачного варианта будет весьма непросто. Он продолжает действовать до тех пор, пока в качестве «горячей клавиши» не будет указано None, причем нужно не просто напечатать это слово, а, поместив курсор в строку, нажать <Shift> + <Backspace>.

А МЫ ОСОБЕННЫЕ!

Выше уже упоминалось о программах, которые отличаются нестандартной работой с памятью и требуют учета некоторых особенностей при создании PIF-файлов. Теперь приведем практический пример.

Под управлением Windows программа может не получить максимально возможного объема отображаемой памяти даже тогда, когда это явно указано в PIF-файле. Дело в том, что существует несколько видов отображаемой памяти. Давным-давно фирмы Lotus, Intel и Microsoft разработали этот тип памяти, чтобы программы могли работать с более объемными электронными таблицами, причем сама отображаемая па-

мять должна была располагаться в адресном пространстве выше 640 Кбайт. Данный стандарт получил название LIM EMS 3.2.

Расширяя возможности LIM 3.2, компании AST Research, Ashton-Tate, Quadram и др. разработали стандарт, допускающий расположение отображаемой памяти как выше, так и ниже 640 Кбайт. Он получил название EEMS (Extended EMS).

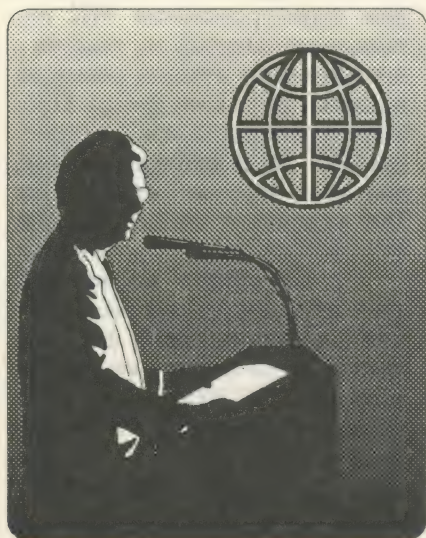
Со временем Lotus, Intel и Microsoft осознали недостатки своего метода и создали усовершенствованный вариант LIM EMS 4.0 по образцу EEMS. Именно его и использует Windows при работе с отображаемой памятью. Между тем, многие приложения, подобно версии 2.01 пакета Lotus 1-2-3, пользуются стандартом LIM 3.2 и «не замечают» страницы отображаемой памяти, расположенные ниже 640 Кбайт. Насколько велики потери памяти в этом случае?

При запуске пакета Lotus 1-2-3 на ПК с процессором 80386 и 4-Мбайт ОЗУ использовался PIF-файл с указанными в нем значениями 816 для KB Required и -1 для KB Limit, однако программе было доступно лишь 376 648 байт отображаемой памяти вместо 816 Кбайт. После этого из памяти были удалены другие DOS-программы, а значение KB Required увеличено до 1280. В результате Lotus 1-2-3 получил 851 552 байта отображаемой памяти — как и в первом случае, программа не смогла обнаружить около 440 Кбайт.

С версией 3.1 того же пакета связана проблема иного рода — в расширенном режиме он способен работать всего лишь с 64 Кбайт расширенной памяти, «добирая» необходимый объем с помощью виртуальной памяти (долгих вам часов у экрана!) или путем нарушения некоторых соглашений Windows об использовании памяти. Фирма, проводившая описанные тесты, утверждает, что пакет Lotus 1-2-3 «едва-едва совместим» с Windows, и считает нецелесообразным запускать его в сеансе DOS.

О ЧЕМ МОЛЧИТ _DEFAULT.PIF?

В заключение хотелось бы напомнить о том, что «злокозненный» _DEFAULT.PIF, тайный враг производительности ваших DOS-программ, по-прежнему пребывает в исходном состоянии. Обратите внимание на рисунки в начале статьи: показанные на них значения параметров соответствуют рекомендуемой установке по умолчанию, позволяющей получить близкую к максимальной производительность и обеспечить надежную работу «среднестатистической» DOS-программы. Откройте файл _DEFAULT.PIF и введите новые параметры, руководствуясь рисунками и внимательно проверяя правильность всех установок. Так как содержимое файла не будет сохранено, если не указать имя существующего файла в строке Program Filename, используйте в качестве заполнителя C:\COMMAND.COM. Завершив работу, измените атрибуты пиктограммы редактора PIF-файлов, задав в командной строке: PIFEDIT _DEFAULT. Атрибут «Только для чтения» защитит _DEFAULT.PIF от случайных изменений. ◇



Итоги конкурса «KEYMACRO.LEX—92»

Г.В. Шмерлинг

Благодаря хитроумным
макрокомандам
текстовый редактор
ЛЕКСИКОН приобретает
новые, удивительно
мощные возможности,
вполне
соответствующие его
огромной популярности
как в России, так и за ее
рубежами.

Призыв сделать вклад в копилку макрокоманд текстового редактора ЛЕКСИКОН, приняв участие в конкурсе, объявленном в журнале «Мир ПК», № 1/92 г., нашел путь к сердцам читателей. Итоги конкурса проводились на прошедшем в рамках выставки «SOFTOOL-92» семинаре СП МИКРОИНФОРМ. К этому времени наша копилка превратилась в увесистую кубышку, полную «меди, серебра и ассигнаций» — в соответствии с уровнем трюкачества в макрокомандах», как написали авторы целого набора команд отец и сын Дунины.

Обладателями призов — врученных Е.Н. Веселовым последних версий редактора ЛЕКСИКОН — стали 12 участников конкурса (было принято решение не ранжировать работы по местам). Кроме того, все конкурсанты стали зарегистрированными пользователями редактора ЛЕКСИКОН (многие письма предварялись честным признанием с типичной формулировкой «... работаю на копии, которую достал по случаю, уж не помню где...»).

При всем разнообразии среди присланных макрокоманд

(МК) лидировали три основные темы: псевдографические конструкции, печать, переопределение МК в сеансе работы с редактором ЛЕКСИКОН.

Приведенные ниже тексты МК проверены в редакторе ЛЕКСИКОН версии 1.1 (8.97).

Набор из четырех выполняемых последовательно МК, позволяющий изобразить прямоугольную рамку произвольного размера, представил В.М. Дворкин.

1. Рисование квадрата:

```
{LEFT}{LEFT}{LEFT}{DOWN}
{LEFT}{LEFT}{UP}
```

2. Увеличение рамки по горизонтали вправо:

```
{LEFT}{DOWN}{LEFT}{UP}
```

3. Уменьшение по горизонтали:

```
{DEL}{DOWN}{DEL}{UP}
```

4. Увеличение по вертикали вниз:

```
{CTRL-RIGHT}{ENTER}{UP}{CTRL-
RIGHT}{DOWN}{UP}
{CTRL-RIGHT}{DOWN}{LEFT}{CTRL-
LEFT}{UP}{RIGHT}
```


Макрокоманду, позволяющую заключить в рамку строку, на которой находится курсор, предлагает Н.И. Нилова. Перед выполнением этой МК следует включить режим замены при вводе символов; длина «заготовленных» в МК горизонтальных линий рамки должна превышать максимальную длину обрамляемых строк:

```
{F3}{CTRL-F3}{CTRL-F4}{CTRL-
F4}{CTRL-F4}{F4}{DOWN}
{HOME}{LEFT}|{LEFT}{UP}
```

```
{DOWN}{HOME}{DOWN}
{UP}{END}
|{LEFT}{UP}|{DOWN}{DOWN}{LEFT}|
{SHIFT-F3}{UP}{UP}{END}
{CTRL-F3}{HOME}
```

К.С. Ахметов и А.В. Пупышев — авторы способа переключения клавиш ввода псевдографики с одинарной рамки на двойную (и вообще переключения заранее определенных наборов МК непосредственно в ходе сеанса работы).

К.С. Ахметов пишет: «Трудно переоценить удобство переопределения МК: нажав <ALT>-F, вы получаете привычный набор конструкций Фортрана, <ALT>-D — dBase, <ALT>-L — любимый набор словосочетаний для деловых писем и т.д.» Для этого некоторым макроклавишам отводится роль переключателей — «менеджеров макрокоманд». С ними связаны МК вида:

```
{F10}тм<буква>{ENTER}<МК>
{ENTER}
```

где <буква> — переопределяемая макроклавиша, <МК> — новый текст МК.

Ниже представлен пример команд, переключающих макросы ввода элементов рамок на одиночные (МАКРО Л) и двойные (МАКРО Д) линии (приведено определение лишь для двух «уголков»):

МАКРО Л

```
{F10}тмй{ENTER}|{ENTER}
{F10}тмц{ENTER}|{ENTER}
```

МАКРО Д

```
{F10}тмй{ENTER}|{ENTER}
{F10}тмц{ENTER}|{ENTER}
```

Отметим, что такой способ применим только для однострочных МК. При попытке переопределить записанную в нескольких строках МК изменится лишь первая строка.

Другие способы переопределения МК основаны на загрузке текстов команд во время сеанса без их записи в файл KEYMACRO.LEX. Так, В.И. Артемьев помещает файл с текстами МК в одно из рабочих окон, причем сами МК располагаются за правой границей окна, оставляя на виду лишь меню из стоящих в начале строк названий. Границу раздела названий и текстов МК следует отметить установкой табулятора.

Процедура работы с меню МК проста: с помощью курсора выбрать по названию необходимую команду и нажать <ALT>-M для ее определения. При этом выделенный из строки макрос окажется связан с буквой V (Variable) до следующей замены.

MACRO Macro — Определение переменной макрокоманды

```
{TAB}
{SHIFT-F3}{END}{CTRL-F3}
{SHIFT-F4}{F4}
{SHIFT-LEFT}
{F10}тмV{ENTER}
{END}{SHIFT-BACKSPACE}
{SHIFT-F4}{F4}{ENTER}
```

Обратите внимание на способ прокомментировать назначение макроса! Следующая за буквой определяемой клавиши часть строки не считывается редактором ЛЕКСИКОН и может служить для комментария.

С.М. Дунин и М.С. Дунин

для переопределения МК используют загрузку их текстов с диска; выбор нужной команды делается по имени файла. Предусмотрена возможность замены МК из нескольких строк, для чего прежнее содержание МК удаляется серией команд {F3}{CTRL-F3} в строке ввода.

Нельзя пройти мимо идеи этих же авторов, весьма полезной для «укрепления дисциплины» среди пользователей редактора ЛЕКСИКОН: автоматическое помещение в начало обрабатываемого документа команд режима деления на страницы, а вместе с ними кратких комментариев к самому документу. Эти комментарии не будут выводиться на принтер при распечатке документа, так как ЛЕКСИКОН игнорирует часть строки, следующую за управляющей командой.

Наберите в первой строке пустого окна имя нового файла и выполните МК, которой отныне следует начинать создание всех документов:

```
{F3}{CTRL-F3}{F10}тс{CTRL-
F4}{F4}{ENTER}{SHIFT-LEFT}
<255>Ш1.5 Содержание:
{ENTER}{SHIFT-LEFT}
<255>В60 Автор :{ENTER}{SHIFT-
LEFT}
<255>Н1 Дата :199 г. Версия:
{SHIFT-UP}{END}{RIGHT}
```

Разумеется, страничный режим и выводимые заголовки можно настроить по своему усмотрению. Вместо числа 255 в угловых скобках следует ввести код 255 (в графических шрифтах редактора ЛЕКСИКОН — инверсный знак вопроса). Если убрать две первые строки МК, имя файла будет запрашиваться в обычном порядке — при первом сохранении текста.

Г.Г. Владимиров предложил МК для подготовки текста к печати по 4 «четвертных» страницы на каждой стороне листа формата А4 (используется ре-

жим принтера condensed — сжатая печать). Из таких листов можно затем изготовить небольшую книжечку.

Приводим фрагмент МК, выполняющий разделение страниц находящегося в окне 1 текста на нечетные и четные, переправляемые в окна 2 и 3 соответственно.

```
{ALT-1}{CTRL-PGDN}{CTRL-PGDN}
{DOWN}{F3}{CTRL-PGDN}{CTRL-F3}
{SHIFT-UP}{SHIFT-F4}{F4}{CTRL-
PGDN}{DOWN}{F3}
{CTRL-PGDN}{CTRL-F3}
{SHIFT-F4}{F4}{F3}{CTRL-
PGDN}{CTRL-F3}
{ALT-3}{SHIFT-DOWN}{CTRL-F4}{F4}
{ALT-1}{CTRL-PGUP}{CTRL-
PGUP}{F3}{CTRL-PGDN}{CTRL-F3}
{ALT-2}{SHIFT-DOWN}{CTRL-F4}{F4}
```

Страницы будут размещены по окнам в следующем порядке:

Окно 2: стр.3 стр.1
 стр.7 стр.5
 ...

Окно 3: стр.2 стр.4
 стр.6 стр.8
 ...

После этого нужно удалить каждый второй разделитель страниц, вставив на его место по 4 пустых строки, и можно печатать текст.

Не приводя тексты команд (попробуйте написать свои варианты!), отметим еще несколько интересных идей.

В.А. Кожанов реализовал с помощью МК... масштабирование шрифта! Во всяком случае, в сторону увеличения — набранный текст можно распечатать большими символами, подготовив, например, плакаты для публичного выступления. Текст в рабочем окне должен быть нарезан на полосы такой ширины, чтобы они могли быть распечатаны на принтере после увеличения букв.

В другое окно помещается шрифтовой файл с выполненными с помощью символов псевдографики рисунками букв, причем каждая матрица буквы снабжена одноименным ключом для поиска. Теперь в дело вступает МК, выполняющая раздвижку строк и поочередную замену символов на соответствующие им матрицы из окна со шрифтом.

А.А. Сердюк создает в одном из окон буфер для фрагментов, расширяющий возможности работы с «карманом» редактора ЛЕКСИКОН.

В.И. Артемьевым с помощью МК и настройки редактора ЛЕКСИКОН при запуске реализована рабочая среда с оконным интерфейсом для ведения иерархической базы документов, их выбора и загрузки для обработки. Нельзя не согласиться с его выводом: использование МК в комплексе с другими возможностями дает удивительные результаты даже при скудном макроязыке.

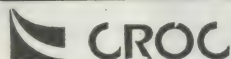
В заключение — маленький сюрприз для любителей макрокоманд: в вашем распоряжении есть ключевое слово {KEY} или {КЛАВИША}, при вставке которого в текст макроса ЛЕКСИКОН ждет нажатия символической или функциональной клавиши, вставляя ее в выполняемую МК. Имеет ЛЕКСИКОН и специальный код для строк комментариев: <255>K (русское) или <255>C (латинское). Эти особенности пока не отражены в документации, и авторы обещают исправить недосмотр в следующем издании.

Готовящийся к выпуску ЛЕКСИКОН 1.2 (8.98) будет содержать и куда более интересные новшества, среди которых поддержка «мыши», пропорциональных шрифтов и многоколонной печати, а также предварительный просмотр страниц (preview) и перезагрузка принтерных драйверов.

Новый конкурс для знатоков редактора ЛЕКСИКОН финансирует 1 февраля 1993 г. На этот раз главным призом, но уже для зарегистрированных пользователей будет компьютер. СП МИКРОИНФОРМ предполагает проводить такие конкурсы и дальше.

ОБ АВТОРЕ

Григорий Владимирович Шмерлинг — старший научный сотрудник СП МИКРОИНФОРМ. Контактный телефон: (095) 233-00-06.



НОВЕЙШИЕ КОМПЬЮТЕРЫ ПРОИЗВОДСТВА США

Фирма «Croc»

поставщик компьютерного оборудования и информационных технологий ведущих производителей США,

предлагает:

- ☐ компьютеры AT-486DX 33, 50, 66 МГц (ISA/EISA);
- ☐ компьютеры 386/486 класса Notebook (в том числе цветные!), портативная периферия;
- ☐ дисковые накопители и подсистемы емкостью 120-2600 Мбайт, QIC/DAT стримеры, КЭШ-контроллеры IDE/SCSI;
- ☐ CD-ROM, WORM, магнитооптика, кассетные накопители BERNOLLI/SyDOS;
- ☐ отказоустойчивые файл-серверы и оборудование для локальных сетей;
- ☐ модемы, факс-модемы и факс-серверы, мультимплексоры и интеллектуальные многоканальные контроллеры ввода/вывода;

- ☐ лазерные принтеры 300-1200dpi, PostScript, цвет;
- ☐ цветные и ч/б сканеры 300-1200dpi;
- ☐ мониторы 17"-21" для издательских систем и САПР, графические ускорители (Windows) и процессоры;
- ☐ источники бесперебойного питания 150-2500 Вт;
- ☐ прикладное и системное программное обеспечение в среде DOS, WINDOWS, OS/2, UNIX, NETWARE.

Поставка оборудования по спецификациям заказчика.
Консультации, сервис, сопровождение.

Телефон в Москве: (095) 200-16-96, факс: (095) 299-46-25, E-Mail: croc@gen.msk.su

ЧЕРНЫЕ ВЕСТИ О ЧЕРНОМ МСТИТЕЛЕ

Брайан Бранденбург



На всякий вирус можно написать антивирус.

Вопрос лишь в том, сколько сил будет потрачено на его разработку, какого он потребует объема памяти и с какой скоростью будет работать.

Из-за Черного Мстителя нам, может быть, скоро придется вывешивать на компьютерах табличку: «10.00 — 14.00: антивирусная профилактика».

Черный Мститель — не герой комиксов, а...

Интересно, каким должен быть человек, который может проводить многие часы за компьютером, не имея ни малейшего шанса на финансовое вознаграждение за свой труд, с единственной целью нанести ущерб тысячам ни в чем не повинных пользователей ПК? А таковой, как выяснилось, существует: дело в том, что автором многих вирусов, включая «Eddie», «Number of the Beast», «Phoenix», «Anthrax» и «Nomenklatura» (последний поразил однажды компьютеры библиотеки Палаты общин), является, по-видимому, один и тот же человек, называющий себя «Dark Avenger» (Черный Мститель). Он претендует также на сомнительную честь считаться автором генератора мутаций — наиболее сложной из известных на сегодняшний день технологий, используемых при создании компьютерных вирусов.

По язвительным замечаниям, вставляемым Черным Мстителем в коды вирусов, можно судить о некоторых его пристрастиях. Так, он определенно равнодушен к «рок-металлу», в частности, к группе «Iron Maiden». Строку «Eddie lives...», по которой получил

свое название вирус «Eddie», большинство экспертов склонны связывать с изображением стилизованной под череп маски «Эдди», используемой «Iron Maiden» в своих шоу-программах. «Anthrax» («Карбункул») — еще одна популярная американская «металлическая» группа. В кодах вирусов содержатся и названия песен «Iron Maiden»: «Somewhere in Time» («Где-то во времени»), «The Evil That Men Do» («Зло, которое совершают люди»), «The Good Die Young» («Хорошие умирают молодыми») и «Number of the Beast» («Число Зверя»). На случай, если кто-либо незнаком с Апокалипсисом, Черный Мститель решил уточнить значение «числа Зверя» и записал в код вируса три байта — «666».

Черный Мститель оценивает себя как наиболее выдающегося автора вирусов в «компьютерном андерграунде». В этом качестве он успел приобрести дурную славу и, видимо, претендует на то, чтобы стать объектом некоего культа. Его вирусы относятся к числу самых сложных и самых разрушительных.

Но Мститель — еще и жертва своего весьма раннего «я»: он очень болезненно реагирует на критику. Когда в марте 1992 г. этот создатель самых зловредных в мире вирусов впервые объявил через международную телекоммуникационную сеть FidoNet о своем генераторе му-

таций и получил в ответ ряд далеко не лестных отзывов, он разразился бранью, после чего куда-то пропал.

Многие, конечно, хотели бы узнать, кто же этот Черный Мститель на самом деле. Используя дедуктивный метод Шерлока Холмса, эксперты на основе высказываний, включаемых Черным Мстителем в код вирусов, и текстов, которые он помещал в электронные доски объявлений, составили «фоторобот» нашего «героя». Под описание, как они утверждают, очень хорошо подходит некий студент-физик, обучающийся в Софийском университете в Болгарии и параллельно работающий в небольшой местной программистской фирме.

Подтвердится ли это предположение? И если да, то понесет ли разоблаченный автор вирусов заслуженное наказание — ведь причиненный его творениями ущерб исчисляется сотнями тысяч долларов? Наиболее вероятный ответ на оба вопроса отрицательный, ибо в Болгарии, как и во многих других странах, нет закона, позволяющего привлечь человека к ответственности за написание компьютерных вирусов. И значит, до тех пор, пока будут существовать типы, подобные Черному Мстителю, нам придется искать способы защиты от их разрушительных изделий.

Прокомментировать статью Брайана Бранденбурга и рассказать о том, что представляет собой генератор мутаций, мы попросили одного из ведущих отечественных специалистов в области компьютерной вирусологии Евгения Валентиновича Касперского.

Человек, подписывающийся как «Dark Avenger», поместил подробный рассказ о своей деятельности по созданию вирусов в болгарские BBS: мол, такого-то числа я написал такой-то вирус, и сделал это по таким-то причинам, тогда-то внес изменения, но данный вариант

распространять не стал, а другую версию отправил туда-то — и так далее в том же духе. Вирусы, которые перечислены в этом своеобразном отчете, совершенно различны, но видно, что «почерк» — один и тот же. Поэтому нет причин сомневаться в том, что Черный Мститель действительно является их автором. Некоторые из его вирусов существуют заведомо более четырех лет, так что если он — студент, то начал их писать чуть ли не в школе.

Этот Черный Мститель, безусловно, талантлив и трудолюбив. Он очень хороший программист и порой находит ин-

тереснейшие решения. Таких людей во всем мире, может быть, десяток. Я понимаю, что (оставим в стороне моральный аспект проблемы) никогда не смог бы написать подобную программу — хотя и считаю себя не самым плохим программистом. И из моих знакомых, среди которых есть отличные программисты, этого сделать, пожалуй, тоже никто не сможет.

Кстати, Черный Мститель занял первое место на одном из заочных конкурсов программистов и каждое задание выполнял к тому же в нескольких вариантах — для ассемблеров разных процессоров.



Все авторы вирусов, которых я видел, насколько можно судить, страдали чем-то вроде комплекса неполноценности. Мститель, видимо, не является исключением, несмотря на весь свой талант. И мне не было бы интересно с ним познакомиться.

Генератор мутаций впервые появился в вирусе, получившем условное название Rogue. Черный Мститель не объявляет себя его автором и, вероятно, таковым не является. Но совершенно точно, что именно он выделил из Rogue модуль, от-

ветственный за мутации, и оформил его в виде объектного файла, который под именем MTE.OBJ (MTE — сокращение от Mutation Engine) вместе с инструкцией по применению поместил в BBS. Теперь любой автор вируса, даже очень средних способностей, может превратить свое сколь угодно примитивное изделие в страшного мутанта. Для этого не нужно ничего придумывать — достаточно скомпоновать вирус с MTE.OBJ, пользуясь инструкцией, простой и всем понятной. Сам вирус может

быть резидентным или нерезидентным, поражать только COM- или EXE-файлы либо и те и другие — короче, возможны абсолютно любые варианты, и многие из них уже реализованы.

Попробую объяснить, что представляет собой генератор мутаций. Обычный вирус имеет всегда один и тот же код, так что мы можем выделить из этого кода характерный фрагмент — маску вируса, после чего по наличию маски определять пораженные файлы и «лечить» их.

«ЖЕЛЕЗНАЯ» ЗАЩИТА

До последнего времени против компьютерных вирусов применялись в основном программные средства, но недавно голландская компания ESaSS предложила им альтернативу в виде специальной антивирусной платы, получившей название Thunderbyte. Главным дистрибутором этих плат является фирма NOVIX (г. Ниймеген, Нидерланды). Гвидо Эррой, генеральный директор фирмы DATA5, имеющей исключительное право на распространение плат Thunderbyte в Германии, считает главным преимуществом антивирусных аппаратных средств то, что они, в отличие от программ-антивирусов, начинают работать еще до загрузки операционной системы и, следовательно, способны отслеживать и обезвреживать в момент активизации даже загрузочные вирусы. Антивирусная программа Thunderbyte записана в СППЗУ и не может быть модифицирована. Она загружается в качестве расширения BIOS раньше каких бы то ни было дру-

гих программ и занимает 1 Кбайт ОЗУ. Сама плата устанавливается в восьмьбитовый разъем шины ISA, EISA или MCA. Действие же ее заключается в том, что она постоянно следит за проявлениями любой специфической активности, характерной для компьютерных вирусов. К таким проявлениям относятся:

- загрузка резидентных программ;
- форматирование жесткого или гибкого диска;
- модификация сектора начальной загрузки;
- модификация таблицы разделов (partition table) жесткого диска;
- перенастройка векторов прерываний;
- снятие у файлов атрибута read-only;
- изменение начальных команд программ;
- добавление фрагментов кода к существующим программам;
- изменение даты создания файлов;
- изменение программ с использованием функций FCB (File Control Block);

- переименование программных файлов;
- переименование файлов, имеющих атрибут read-only;
- прямая запись на диск без обращения к операционной системе.

Кроме того, плата Thunderbyte

- контролирует корректность использования внутренних функций DOS;
- следит за адресами перехода в командах jump;
- защищает от изменений код программ DOS, находящихся в оперативной памяти;
- предохраняет от стирания файлы, каталоги и диски;
- защищает от изменений параметры настройки в CMOS-памяти;
- перед загрузкой системы проверяет сектор начальной загрузки на наличие в нем опасных подпрограмм.

Поскольку плата Thunderbyte ориентирована на распознавание не самих вирусов, а произво-

димых ими действий, для нее не имеет значения тип вируса, она не нуждается в модификации при обнаружении каждого нового вируса и не требует дополнительного времени на сканирование дисков. Как утверждают разработчики, созданная ими плата обеспечивает защиту, которую не сможет преодолеть ни один вирус. К. Сметсма, директор Philips Security Office (г. Эйндховен, Нидерланды), протестировавший большое число антивирусных продуктов, полностью с ними согласен: «Оценив антивирусную плату Thunderbyte с технической и функциональной точек зрения, мы пришли к выводу, что это — оптимальное средство против заражения как существующими вирусами, так и теми, которые могут появиться в будущем». Цена платы Thunderbyte в Германии — 398 немецких марок (285 долларов США).

*Герхард Кафка,
European Computer Sources,
November 1992.*

Довольно долго все так и происходило: для каждого очередного вируса находили маску, добавляли ее к списку таких же масок других вирусов — и антивирусная программа начинала этот вирус распознавать. Но потом появились зашифрованные вирусы, которые по маске искать невозможно: два экземпляра одного вируса, если они зашифрованы с разными ключами, не будут иметь ни одного совпадающего байта — правда, за исключением фрагмента кода, ответственного за расшифровку, который всегда один и тот же. Этот расшифровщик можно найти, по нему расшифровать все остальное и дальше действовать примерно так же, как с обычным вирусом.

В 1991 г. стали появляться вирусы с очень короткими расшифровщиками и с расшифровщиками, состоящими из разных команд. Например, маска зашифровки в одном варианте находится в регистре AX, в другом — в регистре BX, счетчик в одном варианте — в регистре CX, в другом — в регистре SI и т.д. В результате получается расшифровщик, не имеющий в своем коде ни одного постоянного байта. Но можно было «цепляться» за отдельные биты: хотя регистры разные, коды команд все-таки одни и те же. Затем появились расшифровщики, «разбавленные» случайным образом командами, не вызывающими никакого эффекта (например, MOV AX,AX, AND CX,CX, NOP и т.д.), и потребовалось умение «вылавливать» среди таких «пустых» команд содержательные.

В генераторе мутаций МТЕ идея постоянно менять расшифровщик доведена до очень

ВИРУСОЛОГИ ВСЕХ СТРАН, СОЕДИНЯЙТЕСЬ!

По сообщению журнала *European Computer Sources*, вирусологический центр Micro-Bit при Университете г. Карлсруэ (ФРГ) заинтересован в международном обмене опытом. Он планирует опубликовать информацию по 1100 известным в настоящее время вирусам и в дальнейшем регулярно обновлять ее. Контактный телефон центра — (49) 721-376422.

высокой степени совершенства. Если в прежних расшифровщиках могли встречаться, скажем, два десятка различных команд, то МТЕ использует больше половины всех команд процессора, и «пустышек» среди них нет. Длина генерируемого расшифровщика — от 0 до 512 байт, причем вирус может оказаться незашифрованным и при ненулевой длине: все зависит от значений ключа и счетчика, сгенерированных последовательно стью команд, которая каждый раз другая.

Расшифровщик расшифровывает вирус, который после этого копирует свой код в оперативную память. Вирус, находящийся в памяти, не зашифрован, но в момент поражения файла он вызывает модуль генерации зашифровщика и соответствующего ему расшифровщика (это и есть МТЕ), а затем — сгенерированный этим модулем зашифровщик. Зашифровщик шифрует код вируса, приписывает его к расшифровщику, и вирус внедряет полученный код в файл.

Бороться с МТЕ-вирусами очень тяжело. Для сравнения приведу такие цифры: чтобы разобраться с обычным «среднестатистическим» вирусом, т.е. создать программу, которая

бы его обнаруживала и «вылечивала», мне нужно от пяти минут до двух часов, а на вирус с генератором мутаций потребовалось полторы недели.

Моя программа Anti-MTE, которая определяет и «лечит» файлы, пораженные МТЕ-вирусами (она не является самостоятельной, а представляет собой один из модулей большой антивирусной программы), пытается найти в файле расшифровщик, сгенерированный МТЕ, а когда находит — просто расшифровывает тело вируса и записывает его в специальный массив. И с этим массивом уже работает основная антивирусная программа.

Такая процедура, естественно, требует большого объема вычислений и поэтому выполняется медленно, что видно невооруженным глазом: если обычные вирусы удаляются с точки зрения пользователя мгновенно, то, встречая МТЕ-вирус, программа вполне ощутимо «задумывается». И когда пораженных программ много, «лечение» может занять не один час.

Поскольку Черный Мститель сделал генератор мутаций общедоступным, появление большого количества вирусов, основанных на технологии МТЕ, — угроза более чем реальная. Пока мне известны восемь таких вирусов, но это, видимо, только начало...

ОБ АВТОРАХ

Брайан Бранденбург — сотрудник фирмы Symantec, работает в группе Питера Нортон.

Евгений Валентинович Касперский — руководитель отдела в АО «КАМИ», автор книги «Компьютерные вирусы в MS-DOS» (М.: Эдэль, 1992) и разработчик пакета AntiVirus Toolkit Pro.

Любители и профессионалы

И.Б. Рогожкин

Учитывая актуальность темы, редакция журнала «Мир ПК» предлагает вниманию читателей первую из серии статей, посвященных проблемам ремонта и сервисного обслуживания ПК.

За последние несколько лет компьютерный рынок России претерпел существенные изменения. Появилось множество новых фирм, новых продуктов. Изменился и рынок сервисных услуг.

ЛЮБИТЕЛИ

Три года назад я подрабатывал ремонтом ПК. По вечерам, после основной работы, заходил в клуб, где в одной из комнат располагалось малое предприятие, занимающееся ремонтом компьютеров. Все столы в небольшом помещении были заставлены полуразобранными неисправными системными блоками. Вооружившись устройством для прозвонки цепей, осциллографом и паяльником, я начинал свои вечерние бдения. Если неисправность была в системной плате, использовалась утыканная разноцветными светодиодами диагностическая плата, которая вставляется в разъем расширения компьютера и показывает состояние основных сигналов системной шины. Если неисправность проявлялась с прогревом, для ее локализа-

ции использовался маленький «холодильник». При подключении к источнику питания одна сторона этого нехитрого устройства начинает стремительно нагреваться, а другая — охлаждаться. Путем поочередного охлаждения микросхем можно было определить местонахождение неисправности.

Удавалось починить только каждый второй компьютер. Сказывалась нехватка комплектующих и справочной информации. Особо запомнился один случай.

Компьютер AT 286 запустился и нормально работал, но после получасового прогрева «зависал». Привести в нормальное состояние эту машину можно было только дав ей остыть. Методом поочередного охлаждения микросхем мы выяснили, что неисправна программируемая логическая матрица (ПЛМ). Эта микросхема использовалась для формирования тактового сигнала процессора и в зависимости от состояния переключателя Turbo делила частоту кварцевого генератора 48 МГц либо на три, либо на четыре. Она также формировала для процессора сигналы Ready и Reset. «Покопавшись» в справочной литературе, мы нашли отечественный

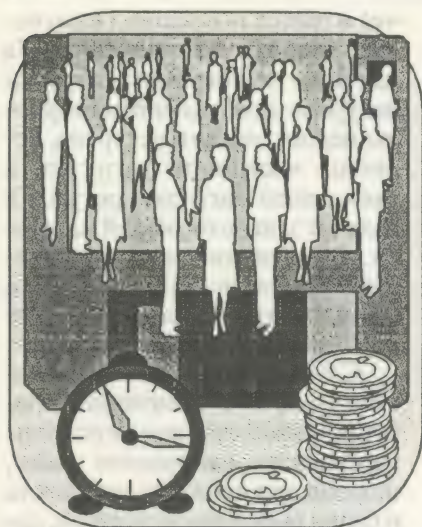
аналог этой микросхемы — 1556ХП6. Однако выяснилось, что время задержки отечественной матрицы составляет 25 нс на вентиль (аналогичной импортной — 7,5 нс). Рассчитывать, что она станет работать на частоте 48 МГц, не приходилось. Получив согласие заказчика на снижение быстродействия компьютера, мы принялись за дело. В поисках программатора для ПЛМ мы вышли на НИИ, где эта микросхема была разработана. Попросили ребят скопировать информацию из иностранной ПЛМ в микросхему 1556ХП6. К счастью, «секретная» перемычка (плавкая перемычка внутри микросхемы, состояние которой разрешает считывание информации) оказалась целой, и никаких проблем с копированием информации не возникло.

Сменив микросхему и кварцевый генератор (с 48 на 35 МГц), мы включили компьютер. Зависать при прогреве он перестал, но работал гораздо медленнее, чем раньше.

Прежде чем закрыть крышку системного блока, мы решили для очистки совести все же попробовать вернуть на место прежний кварцевый генератор. Ура, компьютер заработал! Наши усилия были вознаграждены, но ремонт машины занял целых две недели.

ПРОФЕССИОНАЛЫ

Недавно мне довелось побывать в фирме, которая специализируется на ремонте и техническом обслуживании ПК и сумела захватить половину московского рынка сервисных услуг. Это — Акционерное общество ТЕХНОСЕРВ. В день фирма обслуживает до 50 заказчиков. Здесь установлена локальная сеть, на сервере которой ве-



Time Line 5.0: теперь с графикой!

Сергей Гладков

Новая русская версия системы управления проектами Time Line 5.0 позволяет работать с расписаниями работ неограниченного размера, включает графику, допускает ведение индивидуальных календарей ресурсов, профилирование оплаты и доступности ресурсов, содержит обучающую программу, а также многое другое.

У пользователей русской версии системы управления проектами Time Line 4.0 корпорации Symantec были поводы для недовольства: в ней не было графики, получившей восторженные отзывы в своем оригинальном англоязычном варианте, отсутствовала обучающая программа, не всегда безупречно работала система подготовки отчетов, иногда сбоила система справочных данных... И все-таки это была единственная полностью русифицированная система управления проектами, продающаяся за рубли, и хотя бы поэтому она имела хорошие шансы на успех, которого и достигла благодаря усилиям дистрибуторов.

Русский вариант новой версии, Time Line 5.0, содержит абсолютно все функциональные возможности пятой международной версии и поэтому значительно отличается от предшествовавшей русской версии.

По отзывам пользователей пакета Time Line качество перевода системы — очень хорошее, хотя, конечно, без погрешностей не обошлось. В целом русская версия системы полностью адекватна международ-

ной, а по некоторым параметрам, в частности по оформлению экрана и содержанию справочной системы, превосходит ее.

ТЕХНОЛОГИЯ

Рискну утверждать банальное: вся выгода от использования компьютеров — в повышении эффективности деятельности. Система управления базами данных, издательская система, даже простейший текстовый редактор — все это средства экономии времени, денег, людских, материальных и технических ресурсов. Парадоксально при этом то, что до недавнего времени в России компьютеры для организации управления практически не использовались, а ведь эта область их применения — прямой способ извлечения дополнительной прибыли!

Во всем мире системы управления проектами составляют один из важнейших секторов рынка программного обеспечения. Подсчитано, что выгода от использования таких

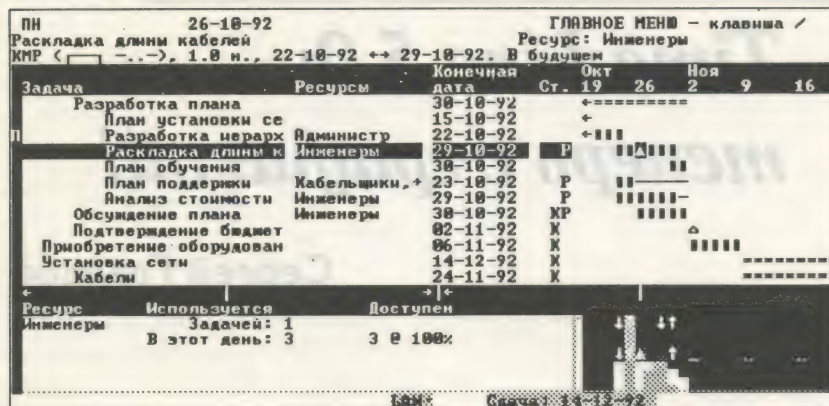


Рис. 1. Основной вид представления данных в Time Line — диаграмма Ганта. На ней отображается список работ (слева) и временная диаграмма их длительности (справа). Одновременно Time Line может показывать загрузку ресурсов в виде гистограммы (справа внизу).

систем — минимум 20% от первоначальной стоимости проекта, и затраты на их приобретение с лихвой могут окупиться на первом же проекте.

Все эти блестящие возможности экономии средств стали доступны отечественным менеджерам после появления на российском рынке русскоязычной версии системы сетевого планирования и управления проектами Time Line 4.0. Пользователями русской версии Time Line 4.0 являются сотни предприятий строительной, машиностроительной, авиационной, атомной и других отраслей промышленности, в том числе ВНИИЖТ, КБ Туполева, КБ Сухого, ЭМЗ им. Ясищева, НПО «Марс», НИИ Радиостроения, АО «Сибирская бумага», НПО «Техномаш», многие другие государственные и частные компании и производственные предприятия.

Новая, пятая версия системы, так же, как и Time Line 4.0, — это средство, предназначенное непосредственно для руководителей — «Лиц, Принимающих Решения» (ЛПР). С помощью этой системы ЛПР может самостоятельно, без привлечения программистов со-

ставить план реализуемого проекта, назначить ресурсы, определить сроки и стоимость выполнения проекта, выявить «узкие места» и заранее устранить их. В результате будут получены реалистичные графики работ и движения денег, установлены сроки поставок. При этом план работ можно улучшить так, что затраты по нему по сравнению с первоначальным вариантом снизятся на 30% и более.

После того как проект разработан, его нужно качественно реализовать. Time Line позволяет осуществлять эффективный текущий контроль за ходом выполнения проекта, оперативно перестраивать его реализацию при возникновении непредвиденных ситуаций — срывах поставок, изменениях графика работы, состава работников, оплаты их труда и т.п.

Time Line 5.0 идеально подходит для нашей действительности, в которой бушуют инфляционные процессы. Для ресурсов (в качестве которых могут выступать как техника и материалы, так и люди) можно определить *профиль доступности* (максимально возможное использование ресурса в зада-

че) и *профиль оплаты* (изменение оплаты во времени). На цветной гистограмме загрузки ресурсов отображаются периоды недоступности ресурсов, периоды частичной загрузки, а также доля загрузки ресурса в данной задаче от общей загрузки. Все эти тонкости учитываются при расчетах временных и стоимостных характеристик расписания выполнения работ.

С помощью Time Line при составлении или изменении проекта можно в считанные секунды пересчитать расписание работ, показав, как влияет на стоимость и сроки выполнения проекта, например, подорожание цемента, уход сотрудника в отпуск или сокращение парка оборудования. Без всякого преувеличения можно сказать, что если бы при разработке графика движения общественного транспорта Москвы использовался Time Line, то уменьшение парка автобусов было бы для москвичей гораздо менее ощутимо.

ВОЗМОЖНОСТИ И ХАРАКТЕРИСТИКИ

Быстродействие

Новая версия Time Line работает примерно в четыре раза быстрее, чем ее предшественница. Увеличена скорость пересчета и построения сетевой диаграммы. Особенно радует скорость загрузки системы — в прежней версии слишком долго демонстрировалась заставка.

Емкость расписаний

В Time Line 5.0 значительно увеличена емкость расписания работ: теперь она определяется только объемом свободной оперативной памяти. При этом Time Line, кроме основной памяти, может использовать

как расширенную, так и отображаемую память. Так, если вы работаете на IBM PC 386 с 4 Мбайт расширенной памяти, то в вашем расписании может быть дополнительно до 4000 задач (кроме 2000, размещаемых в основной памяти); если же в вашем распоряжении имеется IBM PC 486 с 32 Мбайт оперативной памяти, то Time Line одолеет еще 32000 задач (!).

Некоторым пользователям требуется создавать расписания работ столь большого размера, что для них не хватит никакой оперативной памяти. Да и, кроме того, работа со сверхбольшими расписаниями неэффективна. В Time Line и раньше допускалась связь расписаний работ, находящихся в различных файлах, но в версии 5.0 эта возможность значительно улучшена. Теперь, если в расписание работ включаются задачи из «дочернего» расписания, Time Line учитывает загрузку ресурсов в нем при расчете «старшего» расписания работ. Эта возможность практически снимает ограничения на число задач в расписании и позволяет легко структурировать план работ, что особенно важно при работе с Time Line в распределенной сети с нескольких машин, где различные подразделения предприятия могут организовывать работу с помощью своих расписаний.

Модель расчетов

Как и предыдущая версия, Time Line 5.0 допускает неограниченное число уровней вложенности в иерархии расписания, позволяет вести планирование от объема работ. В новой версии алгоритм расчета сетевой модели, и раньше один из самых удачных среди аналогичных систем, значительно усовершенствован. В Time Line 5.0 каждый ресурс может иметь свой собственный календарь,

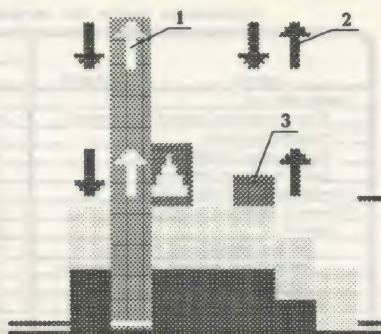


Рис. 2. На гистограмме загрузки ресурсов отображаются все параметры загрузки ресурса в текущей задаче и в расписании в целом (1 — ресурс недоступен, 2 — доступность ресурса ограничена, 3 — перегрузка ресурса).

который учитывается при расчетах. Тем самым в системе реализуется более точная модель расчета, что позволяет поставить эту систему стоимостью около 30000 руб. в один ряд с безумно дорогими пакетами (например, стоимость системы Artemis компании Metier — около 28000 долл.).

В связи с введением индивидуальных календарей ресурсов в Time Line 5.0 реализована уникальная технология *оконтуривания ресурсов* (resource contouring), которая позволяет рассчитывать расписание еще более точно, чем раньше, используя частично назначенные ресурсы, не задействованные в данный момент времени в задачах с более высоким приоритетом.

По вашему желанию Time Line может либо разделить задачу на две части, «обтекающие» период недоступности ресурса, либо задержать ее выполнение до полной доступности ресурса.

Для удобства управления проектом в Time Line 5.0 введены новые расчетные параметры — например, *отрицательный резерв*, показывающий, на сколько было задержано вы-

полнение задачи. Пользователь может также ввести более 20 своих параметров. Новые параметры закрепленных во времени задач позволяют более адекватно моделировать окружающую действительность.

Time Line хранит более 150 различных параметров задач; для расчета сетевой модели этого более чем достаточно. Правда, на предприятии пользователи могут быть приняты свои наименования этих параметров. Однако это не может помешать использовать систему в качестве стандарта предприятия — параметры задач можно переименовывать (например, вы можете называть работников «Персонал», а не «Ресурсы», как это изначально предусмотрено в Time Line).

Макроопределения

В Time Line предусмотрен развитый язык макроопределений, позволяющий, в частности, создавать свои меню и окна диалога. Отдельные макроопределения могут быть вложены друг в друга и все вместе образовывать библиотеку или даже целую программу.

Отмена/повтор команд

Особо следует отметить прекрасно реализованную возможность отмены и повторного выполнения отмененных команд. Time Line, в зависимости от устанавливаемого пользователем верхнего предела, позволяет отменить до 999 последних команд! Это сравнимо по мощности разве что с пакетом AutoCAD.

Импорт/Экспорт

Time Line использует свой собственный формат хранения данных в файлах проектов. Однако в системе предусмотрена не только возможность импорта расписаний работ из старых

версий, в том числе, конечно, и из версии 4.0, но и импорт/экспорт во все наиболее распространенные электронные таблицы и СУБД dBASE.

Коды СРР и ОСРР

Разные задачи в Time Line могут иметь одинаковые имена. При этом для экспорта в электронные таблицы и особенно в базы данных предусмотрена автоматическая система идентификации задач посредством присвоения каждой уникального кода CPP — структуры разделения работ (WBS — Work Breakdown Structure). Кроме того, задачи могут быть идентифицированы по своим характеристикам, например по используемым ресурсам, для чего служат коды OCPP — организационной структуры разделения работ (OBS — Organisation Breakdown Structure). С помощью этих средств расписание работ можно легко преобразовать в реляционную модель данных любой СУБД, например dBASE.

Графика

Но, конечно, самая впечатляющая особенность Time Line 5.0 — это возможность графического вывода диаграмм Ганта и сетевой диаграммы на принтеры и плоттеры.

По результатам опроса пользователей в США, графика Time Line оставляет далеко позади даже графику Microsoft Project (корпорации Microsoft), — системы, полностью использующей возможности графической оболочки Windows.

Графический модуль английской версии Time Line 5.0 имеет свои собственные растровые шрифты, эмулирующие шрифты PostScript, чем и обуславливается высокое качество графической печати. При русификации Time Line 5.0 в

Расписание работ по установке Локальной Вычислительной Сети

[illegible]

При реализации этого проекта необходимо согласование с командантом

Рис. 3. Графический модуль Time Line 5.0 позволяет получить высококачественное представление диаграммы Ганта, которое может быть выведено на лазерном принтере, а на перьевом графопостроителе (плоттере) — даже в цветном варианте.

ПЛАН ВНЕДРЕНИЯ TIME LINE

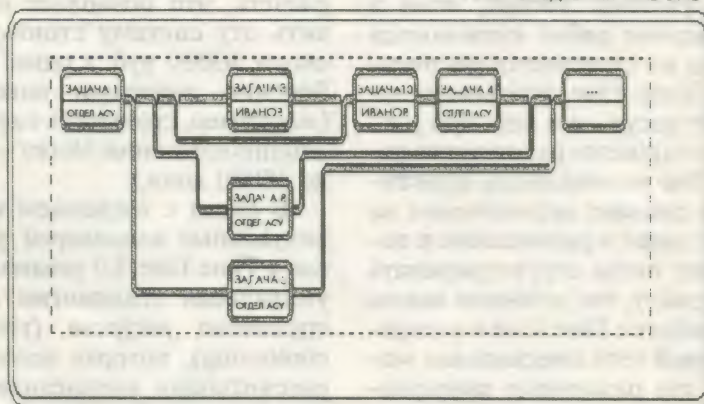
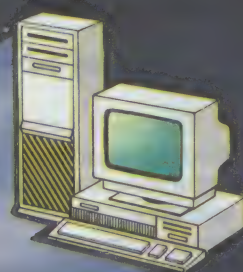
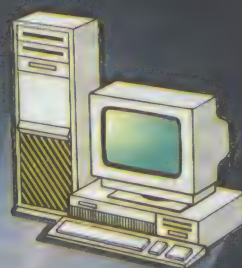
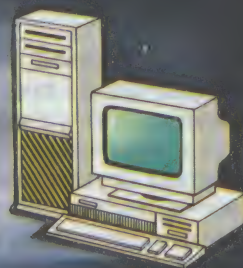


Рис. 4. Сетевая диаграмма проекта, выведенная в графике. Графический модуль Time Line позволяет назначать различным элементам изображения свои цвета. Набором цветов — «Палитрам» — можно присваивать имена и сохранять их на диске.

АО «Диалог-МИФИ» были созданы аналогичные русские растровые шрифты. В результате и русская версия Time Line 5.0 печатает графику так, как будто в вашем распоряжении шрифты и принтер, поддерживающие стандарт PostScript.

Графический модуль Time Line позволяет назначить каждому элементу графика свой

цвет; набору цветов («Палитре») можно присвоить имя и запомнить его. Перед печатью можно просмотреть графическое изображение на экране и, если нужно, изменить его. Time Line позволяет разбить график на полосы и напечатать его на нескольких листах. График можно также автоматически «вписать» в формат листа.



Единственное для Вас!

ЭКСИДЕР®

Работа в локальной сети

Time Line — сетевой продукт; в нем изначально предусмотрена возможность установки и работы в локальной сети. Time Line может осуществлять сквозное регулирование ресурсов по сложным проектам, отдельные части которых физически хранятся на разных машинах в ЛВС, т.е. поддерживает распределенную модель данных. Это и позволяет использовать Time Line как стандарт предприятия, где каждое подразделение может вести свои собственные подпроекты, объединенные в более общий проект, отслеживаемый руководством.

Алгоритм работы Time Line 5.0 в ЛВС несколько изменен: теперь для работы достаточно просто зарегистрировать нового пользователя, сложная поддержка со стороны администратора сети не требуется.

Журнал заметок

В Time Line входит небольшой текстовый редактор, с помощью которого можно комментировать содержание задач и/или ход их выполнения, что очень удобно.

Справочная система

Система справок Time Line полна, достаточна и контекстно зависима: нажатие клавиши F1 вызывает появление экрана, дающего справку о том поле (пункте меню), где при нажатии находился курсор.

Обучающая программа

Одним из недостатков русской версии Time Line 4.0 являлось отсутствие обучающей программы (Tutorial), позволяющей быстро познакомиться с системой, не тратя время на изнурительное изучение 600-страничного руководства пользователя.

КОМПАНИЯ SYMANTEC ВЫПУСКАЕТ TIME LINE ДЛЯ WINDOWS

Корпорация Symantec (Купертино, шт. Калифорния) объявила о выпуске новой системы управления проектами — Time Line для Windows. Тем самым проверенная технология Time Line для DOS полностью переносится в графическую среду Windows. Пользовательский интерфейс Time Line для Windows напоминает электронную таблицу и имеет много сходных черт с интерфейсом системы управления проектами On Target, более раннего и простого своего аналога. По сравнению с версией Time Line 5.0 для DOS, в пакете Time Line для Windows усилена реализация таких функциональных возможностей, как планирование расписания от объ-

ема работ и индивидуальные календари ресурсов. По сравнению с On Target, кроме того, значительно увеличена емкость расписания работ, в котором может теперь быть до 8000 задач.

В версии для Windows реализован метод OLE, позволяющий пользователям объединять информацию из различных источников в одном отчете и обеспечивающий автоматическое обновление отчета при изменении исходной информации. Программа поддерживает пиктограммы, информационные окна, механизм автоматической настройки масштаба, благодаря которому отчет можно расположить целиком на одной странице, а также содержит подси-

стему, помогающую визуально устанавливать причинно-следственные связи между задачами.

Стоимость в США полной версии программы для автономных компьютеров и серверов — 699 долл. Вариант программы для станции ЛВС продается по цене 599 долл. Пользователи Time Line 5.0 для DOS могут приобрести Time Line для Windows по цене 129 долл. Доплата для владельцев пакетов On Target или Microsoft Project равна 199 долл. Так же, как и другие продукты Symantec, систему Time Line для Windows (правда, пока только англоязычную версию) можно приобрести и в России; ее стоимость — 25000 руб.

В Time Line 5.0 включена обучающая программа, созданная в АО «Диалог-МИФИ». Просмотрев ее первые шесть уроков, пользователь может начать работу в Time Line.

Примеры расписаний

В русскую версию Time Line 5.0 включены примеры, близкие к нашей действительности: постановка на производство изделия, план ремонта здания и пр. Эти расписания работ были взяты из реальной практики внедрения Time Line 4.0 на российских предприятиях.

Документация

В комплект поставки русской версии входит полное руководство пользователя объемом 600 страниц, накладка на клавиатуру, справочная брошюра и лицензионное согла-

шение. Эти материалы, особенно руководство пользователя, значительно переработаны по сравнению с оригинальными и хорошо структурированы.

СОПУТСТВУЮЩИЕ ПРОДУКТЫ

Для облегчения знакомства и перехода на Time Line 4.0 в АО «Диалог-МИФИ» была выпущена так называемая учебная версия Time Line. Учебная версия — это произведенный с разрешения корпорации Symantec программный продукт, предназначенный для обучения. В комплект поставки учебной версии входят: дистрибутив Time Line 4.0 с ограничением в 32 задачи, регистрационная карточка и учебное пособие объемом 150 страниц.

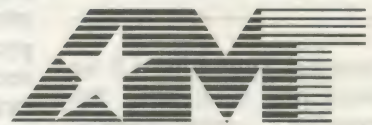
MeDiann Ltd.

IBM-совместимость не мираж, а жестокая реальность для тех из нас, кому стали тесны рамки MS-DOS. Мощные программные продукты - Novell NetWare, Unix, Xenix, OS/2 и даже Windows 3.1 предъявляют гораздо более высокие требования к архитектуре материнских плат и плат расширения. И только лидерам под силу создавать изделия, соответствующие этим требованиям. Мы приводим список игроков нашей команды:

386SX 25MHz
386DX 40MHz
486DX 33MHz (local video bus)
486DX 50MHz (local video bus)
ProDesigner IIs (1024 by 768)
ProDesigner IID (local video bus)
Fahrenheit 1280D (1280 by 1024, S3 graphics coprocessor, local video bus)



модемы



AMT 2442 V.42bis/MNP-5 AMT 2400 (soft MNP)
AMT 9642 V.42bis/MNP-5 AMT 2496 faxmodem
AMT 1442 V.42bis/MNP-5

контроллеры жестких дисков

UltraStor 14F ISA SCSI-2
24F EISA SCSI-2
MYLEX DNE-710 EISA SCSI-2
DCE-376 EISA SCSI-2



MultiMode 120 Dual Page 19" monochrome monitor
PageView Full Page monochrome monitor

FUJITSU
Maxtor

жесткие диски

330MB-1.7GB
(200,000MTBF)
80MB-1.7GB
(150,000MTBF)



Ethernet 16-bit card

MYLEX

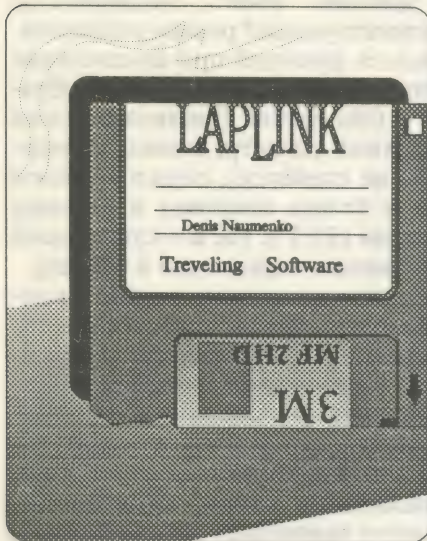
LNE-390A EISA card



*Заказные Конфигурации Компьютеров
Локальные и Распределенные Сети*

*Москва, Петровка 25
тел (095) 200-6020
факс (095) 209-5171*





Общая библиотека для Си и Паскаля

В.А. Нечаев,
А.Н. Малярский

Рассматриваемая
в статье технология
написания библиотечных
модулей, совместимых
с различными системами
программирования
на Си и Паскале,
поможет программистам
расширить список
компиляторов, для
которых предназначены
создаваемые ими
библиотеки,
и сэкономить время,
силы и средства.

При создании проблемно-ориентированных библиотек и других подобных продуктов разработчик, естественно, стремится сделать свой пакет совместимым (хотя бы на уровне исходного текста) с возможно большим числом распространенных систем программирования, причем приложив минимум усилий. Чтобы этого добиться, необходимо использовать соответствующую технологию написания и переноса программ.

Описываемая здесь технология применялась при разработке пакета HyTech — высокоскоростной СУБД ядерного типа с CALL-интерфейсом для языков Си и Паскаль (собственного языка в HyTech нет). Библиотеки HyTech, написанные на Borland C++ 3.0, затем переносились в другие среды программирования на Си (Microsoft C 6.0, Turbo C 2.0, Turbo C++ 1.0 и Borland C++ 2.0) и в систему Turbo Pascal (версии 5.5 и 6.0).

Чтобы читатель получил представление о сравнительной трудоемкости процессов разработки программы и переноса ее в новую среду, упомянем, что перенос библиотеки для работы с диалоговыми панелями (аналог Turbo Vision) занял две недели, в то время как на написание и отладку соответствующих программ потребовалось восемь месяцев (их исходный текст содержит около 50000 строк).

ОБЩИЕ ЗАМЕЧАНИЯ

Результатом трансляции для модуля на языке Си является объектный файл в формате OBJ, а для модуля на Паскале — файл в формате TPU. Совместимость их достижима только за счет того, что при компоновке Паскаль-программ в исполняемый модуль могут, наряду с TPU-файлами, включаться файлы в формате OBJ*.

* В компиляторе Borland Pascal 7.0 появилась возможность создавать динамически подключаемые библиотеки (файлы в формате DLL); эти библиотеки поддерживаются (правда, только для среды Windows) также рядом Си-компиляторов. Однако применение формата DLL ограничено защищенным режимом, работа в котором (во всяком случае, в нашей стране) распространена пока не очень широко.

Таким образом, функции, предназначенные для использования в программах на Си и Паскале, очевидно, должны быть написаны на Си — это позволит получить OBJ-файл, который воспринимает оба компоновщика.

Воспользоваться возможностями языка Си++ (во всяком случае, Borland C++), к сожалению, не удастся: мешает так называемая перегрузка функций (function overloading) — присвоение в программе одного и того же имени нескольким функциям, различающимся по числу или типу параметров. Для ее реализации компилятором используется техника «искажения имен» (name mangling): скажем, функции `int test(char c, int i, long l)` будет присвоено внутреннее имя `@test$zqzcl` (вспомним, что в Turbo Pascal имя функции не должно содержать символов `@` и `$`). «Искажение имен» можно отключить, если описывать функции в программе на Си++ как `extern «C»`, но это — практически то же самое, что просто использовать обычный Си.

Для отладки Паскаль-программы, обращающейся к функциям, реализованным на Си, можно применять отладчик Borland Turbo Debugger, однако все такие функции он позволяет видеть лишь в ассемблерных кодах, поскольку «не подозревает» о существовании их исходных текстов, написанных на другом языке*. Для опытного программиста, вероятно, не составит труда следить за ходом выполнения Си-программы по ее исходному тексту в отдельном окне.

ОГРАНИЧЕНИЯ, ОБЕСПЕЧИВАЮЩИЕ ПЕРЕНОСИМОСТЬ ПРОГРАММ

При написании Си-функции, которую предполагается вызывать из программ на Си и Паскале, необходимо соблюдать следующие правила.

1. Запрещается пользоваться библиотечными функциями, не имеющими точного аналога в библиотеках других компиляторов.

При наличии таких аналогов можно подставить соответствующую функцию для каждого из языковых пакетов, применив так называемую «подмену функций» (эта техника будет рассмотрена более подробно в связи с функциями распределения памяти). Если же хотя бы в одном из поддерживаемых пакетов аналог отсутствует, функцию необходимо написать самостоятельно (или воспользоваться каким-либо готовым исходным текстом).

* Версия Turbo Debugger 3.2, поставляемая с пакетом Turbo Pascal 7.0, позволяет работать с модулями как на Паскале, так и на Си и Ассемблере. — (Прим. ред.)

2. Функции должны иметь модификатор `pascal` (либо соответствующий режим устанавливается сразу для всех функций при компиляции библиотеки). Этот модификатор задает преобразование имени в OBJ-файле (символы имени переводятся в верхний регистр, символ подчеркивания в начале не ставится), порядок записи параметров в стек (слева направо) и команду возврата из функции (`RET p`, где `p` — суммарный размер всех переданных параметров в байтах).

3. Все глобальные переменные следует вынести в отдельный модуль. Кроме того, недопустимы инициализированные статические переменные — вместо них также используются вынесенные в отдельный модуль глобальные переменные, объявляемые как `extern` в тех модулях, где на них имеются ссылки. Инициализация, в том числе и инициализация нулем, которую Си-компиляторы могут выполнять автоматически, должна быть явной.

Модуль глобальных переменных будет участвовать только в компоновке Си-программ; для Паскаля потребуется определить соответствующие переменные в главной программе или отдельном модуле на Паскале.

4. Все литеральные константы необходимо описать в модуле глобальных переменных. Это связано с тем, что в Паскаль-программах соответствующие константы располагаются в кодовом сегменте и, следовательно, должны быть определены в тексте модуля.

Таким образом, Си-функция, содержащая конструкцию вида

```
strcmp(MyString,"PROGRAM");
```

не может быть скомпонована с программой на Паскале, и вместо этой конструкции следует написать:

```
...
extern char ProgramId[ ];
...
strcmp(MyString,ProgramId);
```

включив в модуль глобальных переменных строку

```
char ProgramId[ ]="PROGRAM"
```

5. Не следует пользоваться модификаторами `loadds` и `interrupt`. Функции с этими модификаторами загружают в регистр DS значение сегмента `CONST` (а не `DATA`, как принято в Turbo Pascal), в результате чего компоновщик Turbo Pascal не может с ними работать и выдает сообщение «Invalid relocatable reference» (неверная перемещаемая ссылка).

Листинг 1. Ассемблерная часть функции new_kbd_interrupt.

```

extrn pascal new_kbd_interrupt: far
; Описание функции, реализованной на Си.
proc pascal new_kbd_strategy far
; Обработка прерываний от клавиатуры.
uses ax,bx,cx,dx,es,ds,si,di,bp
; Зададим сохранение регистров.
; Установим сегмент данных для доступа
; к переменным программы, пользуясь
; регистром bp.
IFIDNI MDL, <TPASCAL> ; Для Паскаль-программ:
    mov bp, DATA ; сегмент данных.
ELSE ; Для Си-программ:
    IF @DataSize GT 1
        mov bp, @fardata
        ; Сегмент данных для модели HUGE.
    ELSE
        mov bp, @data
        ; Сегмент данных для других моделей.
    ENDIF
ENDIF
    mov ds, bp.
; Загрузим в ds полученное значение.
call new_kbd_interrupt pascal
; Вызовем "старую" процедуру обработки
; прерывания.
    iret ; Выйдем из обработчика.
endp
public pascal new_kbd_strategy

```

При необходимости включить в библиотеку такого рода функцию ее потребуется переписать. Целесообразно разбить эту функцию на две части — вызывающую (на ассемблере) и вызываемую (на Си), текст которой будет отличаться от первоначального только отсутствием модификаторов `loadds` или `interrupt`.

Пусть, например, нужно включить в программу новый обработчик прерываний от клавиатуры, названный `new_kbd_interrupt`. Тогда его ассемблерная часть может выглядеть так, как показано в листинге 1. Чтобы сгенерировать для нее код, совместимый с Turbo Pascal, необходимо задать транслятору Borland Turbo Assembler переключатель `/dMDL=TPASCAL`, если же требуется совместимость с Си, задается значение `/dMDL=SMALL (LARGE, MEDIUM, COMPACT, HUGE — в зависимости от модели памяти)`.

Встроенным ассемблером (BASM) в данном случае воспользоваться, к сожалению, невозможно: он «не признает» имен сегментов, отличающихся от принятых для компилятора Си, даже если попытаться указать их при компиляции в явном виде.

Для наглядности в листинге 2 представлена обычная программа на языке Си, и она же — в «переносимой» форме.

Листинг 2. Различия между обычной и «переносимой» программой на языке Си.

<pre> /* Исходный текст : */ /* Файл "MY1.C": */ #include <string.h> int i; static int s=8; int my_func(int n, char* s) { i=n+1; s=i-1; if (!strcmp(s,"PROGRAM")) return(n); else return(i); } </pre>	<pre> /* "Переносимая" форма : */ /* Файл "MY1.C": */ /* Реализация strcmp */ int strcmp(const char* s1, const char* s2) { } /* Описания внешних переменных */ extern int i; /* Глобальная */ extern int s; /* описатель static недопустим */ extern char ProgramId []; /* Литеральные константы недопустимы */ int my_func(int n, char* s) { i=n+1; s=i-1; if (!strcmp(s,ProgramId)) return (n); else return(i); } /* Файл "MYVARS.C": */ /* Выделение памяти под переменные и их инициализация */ int i; int s=8; /* ранее неявно инициализировалась нулем, поскольку */ /* была описана как static */ char ProgramId[]="PROGRAM"; /* Инициализация литеральной константы */ </pre>
---	---

КОМПИЛЯЦИЯ СИ-ФУНКЦИЙ ДЛЯ СРЕДЫ TURBO PASCAL

Список опций компилятора Borland C++ 3.0 для компиляции программы, переносимой в среду Turbo Pascal, приведен в таблице. Прокомментируем некоторые из них.

Поскольку переноситься будет объектный модуль, сборка не нужна, достаточно только компиляции (параметр `-c`). Использовать отладочную информацию все равно не удастся, поэтому генерировать ее бессмысленно (опции `-v` и `-y`)*. Исходные модули должны обрабатываться именно транслятором Си, а не Си++ (опция `-P-C`).

Компилятор Turbo Pascal генерирует для данных лишь дальние (`far`) указатели, а для функций

— как ближние (`near`), так и дальние: например, в интерфейсной части модуля все функции по умолчанию дальние, в разделе реализации — ближние. Можно и непосредственно задать адресацию функции с помощью опций `{F+}` (дальняя адресация) и `{F-}` (ближняя адресация). Таким образом, с точки зрения языка Си в Turbo Pascal реализованы две модели памяти — COMPACT (ближние указатели на код, дальние — на данные) и LARGE (дальние указатели и на код, и на данные).

Отсюда следует, что переносимая программа должна быть написана и откомпилирована в модели LARGE или COMPACT, и в обоих случаях можно применять модификаторы `near` и `far` для функций. При разработке HyTech авторы выбрали самый простой и надежный вариант — сделали все функции дальними (для Си-функций всегда устанавливалась модель LARGE, а в модулях Turbo Pascal — опция `{F+}`).

Запрет на использование библиотечных функций относится, естественно, и к функциям,

* Отладочная информация пригодится при работе с Turbo Debugger 3.2 (см. прим. на стр.98).

Таблица. Опции компилятора, обеспечивающие совместимость Си-программы со средой Turbo Pascal.

Файл конфигурации	Интегрированная среда	Комментарий
<code>-c</code>	Options -> Make -> After compiling = Stop	Только компилировать
<code>-ml</code>	Options -> Compiler -> Code generation -> Model = Large	Модель памяти — LARGE
<code>-f</code>	Options -> Compiler -> Advanced code generation -> Floating point = None(*)	Не использовать плавающую точку
<code>-N</code>	Options -> Compiler -> Entry -> Exit Code -> Stack options -> Test stack overflow[off]	Не проверять переполнение стека
<code>-r</code>	Options -> Compiler -> Optimizations -> Register variables = None	Не использовать регистры для хранения переменных
<code>-p</code>	Options -> Compiler -> Entry -> Exit Code -> Calling convention Pascal	Использовать паскалевское соглашение о вызове функций
<code>-v</code>	Options -> Compiler -> Advanced code generation -> Debug info in OBJ[off]	Не генерировать отладочную информацию
<code>-y</code>	Options -> Compiler -> Advanced code generation -> Line numbers debug info[off]	Не генерировать информацию о номерах строк в объектном файле
<code>-P-C</code>	Options -> Compiler -> C++ options -> Use C++ compiler CPP Extension only	Вызывать компилятор языка Си для файлов с расширением C; установить по умолчанию расширение C
<code>-zCCODE</code>	Options -> Compiler -> Names -> Code segment=CODE	Установить имя CODE для сегмента кода
<code>-zP</code>	Options -> Compiler -> Names -> Code group='*'	Группа для кода отсутствует
<code>-zA</code>	Options -> Compiler -> Names -> Code class='*'	Класс для кода отсутствует
<code>-zRCONST</code>	Options -> Compiler -> Names -> Data segment=CONST	Установить название CONST для сегмента данных
<code>-zS</code>	Options -> Compiler -> Names -> Data group='*'	Группа для данных отсутствует
<code>-zT</code>	Options -> Compiler -> Names -> Data class='*'	Класс для данных отсутствует
<code>-zDDATA</code>	Options -> Compiler -> Names -> BSS Segment=DATA	Установить название DATA для сегмента BSS
<code>-zG</code>	Options -> Compiler -> Names -> BSS group='*'	Группа для BSS отсутствует
<code>-zB</code>	Options -> Compiler -> Names -> BSS class='*'	Класс для BSS отсутствует

предназначенным для вычислений с плавающей точкой, поэтому устанавливается опция -f-.

Проверка переполнения стека (опция -n-) отменена, так как она требует доступа к некоторым глобальным переменным, вводимым в стартовом модуле C0x.OBJ (понятно, что в Паскаль-программе их нет).

Применение регистровых переменных (опция -r-) несовместимо с вызовом Паскаль-функций из Си-функций: Си-функция ожидает, что содержимое регистров SI и DI не изменится, а Паскаль-функция об этом «не догадывается». Если вызовов такого типа заведомо не будет, отменять автоматическое использование регистровых переменных не обязательно.

Для правильной работы компоновщика Turbo Pascal необходимо, чтобы кодовый сегмент имел имя CODE, сегмент данных — имя CONST и сегмент BSS (неинициализированных данных) — имя DATA. Это обеспечивают опции, названия которых начинаются с -z.

TURBO PASCAL: ОПИСАНИЕ БИБЛИОТЕКИ И КОМПОНОВКА

Разумно поместить вызовы Си-функций и все функции-оболочки, поддерживающие такие вызовы, в отдельный модуль Turbo Pascal и затем применять его в прикладных программах (структура независимо компилируемого модуля (unit) в Turbo Pascal показана в листинге 3).

Листинг 3. Структура модуля для Turbo Pascal.

```
Unit Имя_модуля;
INTERFACE (* Раздел описаний *)
  Uses Имя_исп_модуля;
  (* Имена используемых модулей *)
  .....
  (* Описания глобальных переменных,
  доступных другим модулям *)
  .....
  (* Описания функций, доступных другим модулям *)
  .....
IMPLEMENTATION (* Раздел реализации *)
  Uses Имя_исп_модуля;
  (* Имена используемых модулей *)
  .....
  (* Описания внутренних функций и переменных *)
  .....
  (* Реализация функций *)
  {$L Имя подключаемого *.OBJ - файла }
  {$L Имя подключаемого *.OBJ - файла }
  .....
  {$L Имя подключаемого *.OBJ - файла }
BEGIN (* Секция инициализации *)
  .....
END.
```

В этом модуле описываются, кроме того, все глобальные переменные и константы (те же, что были выделены в отдельный модуль при написании Си-функций). Инициализация может производиться путем объявления константы:

```
Const MyWord:Word=200;
```

Возможна также явная инициализация переменной в секции инициализации модуля:

```
.....
IMPLEMENTATION
  Var MyWord:Word;
  .....
  Begin
    MyWord:=200;
  End
```

Все функции, не объявленные в Си-модуле с атрибутом static (и, следовательно, доступные для вызова из других программ), должны быть описаны как External в разделе IMPLEMENTATION, причем это касается и тех функций, к которым модуль на Паскале в действительности не обращается. Аналогично, все внешние имена, описанные в Си-модуле как extern, должны быть объявлены посредством предложения {\$L ...}. В противном случае компоновщик Turbo Pascal выдает соответственно сообщения «Invalid public definition...» (недействительное определение общедоступной переменной) и «Invalid EXTRN definition...» (недействительное определение внешней переменной).

РАБОТА С ДИНАМИЧЕСКИ РАСПРЕДЕЛЯЕМОЙ ПАМЯТЬЮ И «ПОДМЕНА ФУНКЦИЙ»

Одной из важнейших задач, возникающих при написании совместно используемого кода, является организация работы с динамически распределяемой памятью.

Функции, обеспечивающие работу с динамически распределяемой памятью, присутствуют в библиотеках всех рассматриваемых здесь пакетов, но структура ее не документируется и может быть различной в зависимости от компилятора и даже от версии компилятора. Поэтому в данном случае уместно применить упомянутый выше принцип «подмены функций»: включить в свою библиотеку функцию-оболочку, которая будет вызывать ту или иную библиотечную функцию в зависимости от предложений #define. Пример такой оболочки для функции farcalloc

Листинг 4. «Подмена функций»: функция farcalloc.

```

/* Описание прототипов функций */

#ifdef FOR_BORLAND /* Для Borland C */
void far* farcalloc(unsigned long nunits,
                    unsigned long unitz);
#else
#ifdef FOR_MSC /* Для Microsoft C */
void far* _fcalloc(unsigned nunits, unsigned unitz);
#else
#ifdef FOR_PASCAL /* Для Turbo Pascal */
void pascal far* PascalFarCalloc (unsigned BlockSize);
#endif
#endif
#endif

.....

/* Пользовательская функция выделения памяти */
/* Не запрашивайте более 64К ! */

void far* MyFarCalloc(unsigned nunits, unsigned unitz)
{
    return
#ifdef FOR_BORLAND
    farcalloc(nunits,unitz)
#else
#ifdef FOR_MSC
    _fcalloc(nunits,unitz)
#else
#ifdef FOR_PASCAL
    PascalFarCalloc ((unsigned) (nunits*unitz))
    ;
    ;
    ;
#endif
#endif
#endif
}
#endif
#endif
#endif

```

приводится в листинге 4. Листинги 5 и 6 содержат вариант реализации на Паскале функций выделения и освобождения памяти — соответственно PascalFarCalloc и PascalFarFree (заметим, что в Turbo Pascal в первых двух байтах блока записывается его длина).

Таким образом, для Borland C вызывается farcalloc, для Microsoft C — _fcalloc, а для Turbo Pascal — PascalFarCalloc. При этом первые две функции имеются в соответствующих библиотеках, а PascalFarCalloc вызывает встроенную в язык Turbo Pascal функцию GetMem.

НАПИСАНИЕ, КОМПИЛЯЦИЯ И ЗАПУСК ПАСКАЛЬ-ПРОГРАММ В СРЕДЕ BORLAND C++

Как это ни парадоксально, но программы на Паскале, использующие библиотеки, написанные на Си, очень удобно писать и отлаживать в среде Borland C++.

Листинг 5. Паскаль-функция, аналогичная Farcalloc в языке Си.

```

(* Функция выделения и очистки блока памяти *)
Function PascalFarCalloc(BytesNeeded:Word);
Type PtrRec = Record
  Ofs, Seg: Word;
  (* Структура для доступа к указателю *)
End;
Var
  P: ^Word; (* Указатель *)
  Len: Word;

Begin
  Len := BytesNeeded + 2;
  (* Запрашиваем на 2 байта больше *)
  GetMem( P, Len ); (* Выделяем память *)
  If P <> Nil Then
    Begin
      (* Память выделена *)
      P^ := Len; (* Запоминаем длину блока *)
      (* в двух первых байтах *)
      Inc( PtrRec(P).Ofs, 2 );
      (* Увеличиваем значение P *)
      FillChar(P^, BytesNeeded, #0);
      (* Заполняем оставшийся *)
      (* участок нулями *)
      PascalFarCalloc := P; (* Возвращаем P *)
    End
  Else PascalFarCalloc := Nil;
  (* Память не выделена; возвращаем Nil *)
End; { PascalFarCalloc }

```

Листинг 6. Функция PascalFarFree.

```

(* Освобождение блока памяти *)
Procedure PascalFarFree(BlockAddr:Pointer);
Type PWord = ^Word;
Type PtrRec = Record
  Ofs, Seg: Word;
  (* Структура для доступа к указателю *)
End;
Var Len: Word;
Begin
  If BlockAddr <> Nil Then
    (* Если адрес не равен Nil, *)
    (* освобождаем память *)
    Begin
      (* Уменьшаем полученное *)
      (* значение указателя *)
      Dec( PtrRec(BlockAddr).Ofs, 2 );
      (* Определяем длину блока *)
      Len := PWord( BlockAddr )^;
      (* Освобождаем блок *)
      (* заданной длины *)
      FreeMem( BlockAddr, Len );
    End;
  End;
End;

```

Чтобы компилировать Паскаль-программу не выходя из среды, достаточно добавить к списку внешних программ Borland C++ транслятор Turbo Pascal (если при добавлении указать, что программа является транслятором, ее можно будет применять при трансляции файлов, входящих в проект).

Для модулей, обрабатываемых компилятором Turbo Pascal, необходимо выбрать этот компилятор из списка в меню *Project -> Local options*.

Дав программе имя, совпадающее с именем проекта, вы сможете выполнить откомпилированную Паскаль-программу из среды Borland C++.

При добавлении внешней программы необходимо указать макросы, связанные с ее выполнением. Опишем макросы, которые могут оказаться полезными при работе с компилятором Turbo Pascal.

\$CAP EDIT — выводить сообщения компилятора в стандартное окно Transfer Output. Это позволяет использовать компиляторы, для которых отсутствуют фильтры сообщений.

В действительности компилятор Turbo Pascal с точки зрения обработки сообщений устроен довольно просто. При компиляции с ключом /Q (Quiet compile — тихая компиляция) он выводит не непрерывно «бегущие» по экрану номера строк, а либо число откомпилированных строк (в случае успеха), либо сообщение об ошибке (единственной!), которое несложно передать в стандартное окно «Message». Информацию о формате сообщения можно почерпнуть из текста стандартных программ-фильтров (GREP2MSG и TASM2MSG). Авторам удалось написать, отладить и подключить фильтр к среде Borland C++ достаточно быстро. Полный текст фильтра приводится в приложении.

\$CAP MSG(TPC2MSG) — задать имя филь-

тра, обеспечивающего обработку сообщений от компилятора Turbo Pascal.

\$EDNAME — компилировать файл, находящийся в текущем окне редактора.

\$NOSWAP — не переключать экран на текущую выполняемую программу. Вместо переключения на экран выводится сообщение «Running Имя_Программы...».

\$MEM(SIZE) — освободить для внешней программы память объемом SIZE. Данный макрос позволяет освободить до 640 Кбайт памяти, однако при компиляции и компоновке больших программ компилятору Turbo Pascal даже этого иногда недостаточно; в таких случаях целесообразно вызывать компилятор с опцией /L (Link buffer to disk — разместить буфер компоновки на диске).

\$SAVECUR — запомнить текущий редактируемый файл перед компиляцией, если он был изменен. Это необходимо сделать, поскольку компилятор Turbo Pascal берет файл с диска, а не из памяти редактора интегрированной среды.

Как видим, Borland C++ и здесь в очередной раз подтверждает свою репутацию очень гибкой и мощной среды, по праву считающейся лучшей средой программирования в настоящее время.

ОБ АВТОРАХ

Вадим Александрович Нечаев и Андрей Николаевич Малайский — программисты, сотрудники фирмы HyTech. Контактный телефон (095) 324-44-07.

Приложение. Текст программы-фильтра TPC2MSG.

```
/*      TPC2MSG.C      */
/* Фильтр для преобразования сообщений от */
/* компилятора Turbo Pascal в интегрированную */
/* среду Borland C++ 3.0      */

/*      Copyright (c) HyTech 1992      */

/* Компилятор Turbo Pascal выдает сообщение об
   ошибке в следующем формате :

Turbo Pascal Version ... Copyright (c) ...\
Borland International
Имя_файла (Номер_строки): Error Номер_ошибки :\
Текст_сообщения
Строка_с_ошибкой
^

Таким образом, необходимо проверить, имеется
ли слово "Error" во второй строке.
*/

#include <io.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "filter.h" /* Заголовочный файл,
```

```
поставляемый вместе с примерами системы
Borland C++ 3.0 */
#define STDOUT 1 /* Номер обработчика stdout */
#define MAXLEN 255
/* Максимальная длина строки при чтении файла */
#define WRITE(x) write(STDOUT,&x,sizeof(x))
/* Макрос для записи в stdout */

void main(void)
{
    char LineRead [MAXLEN]; /* Буфер чтения */
    char FileName [80]; /* Имя файла */
    char* PtrToErrorWord;
        /* Указатель на слово «Error» */
    char MsgType; /* Тип сообщения */
    char LineNumberString [6];
        /* Номер строки с ошибкой (строка) */
    unsigned LineNumberWord;
        /* Номер строки с ошибкой (число) */
    char* AuxPtr;
        /* Вспомогательный указатель */
    char* PtrToLineBeg=LineRead;
        /* Указатель на начало буфера */
    char found=0; /* Флажок */
    int len; /* Длина различных строк */
    int ColumnNo; /* Номер колонки */

    setmode(STDOUT,O_BINARY);
        /* Установить двоичный режим для stdout */
```



```

write(STDOUT,PipeId,PipeLen);
/* Записать в STDOUT признак начала работы
с окном сообщений */
if (!fgets(LineRead,MAXLEN,stdin))
/* Прочитать первую строку - она не нужна */
goto End; /* Прочитать не удалось */

if (!fgets(LineRead,MAXLEN,stdin))
/* Прочитать вторую строку */
goto End; /* Прочитать не удалось */

LineRead[strlen(LineRead)-1]='\x0';
/* Убираем символ '\n' в конце строки */
PtrToErrorWord=strstr(LineRead,": Error");
/* Ищем во второй строке подстроку «Error» */

if (PtrToErrorWord) /* Нашли, продолжаем... */
{
MsgType=MsgNewFile;
/* Идентифицировать сообщение как сообщение
о новом файле */
WRITE(MsgType); /* Записать в stdout */

/* Ищем в строке с ошибкой символ «(», после
которого идет номер строки */

for (AuxPtr=PtrToErrorWord;
#pragma warn -pia
(AuxPtr>=PtrToLineBeg) &&
!(found=(*AuxPtr=='('));
#pragma warn +pia
AuxPtr++)
;

if (found) /* Скобка найдена */
{
len=(int) (AuxPtr-PtrToLineBeg);
/* Длина имени файла */
memcpy(FileName,LineRead,len);
/* Копируем имя файла */
FileName[len]='\x0'; /* Конец строки */

```

```

/* Записываем в stdout */
write(STDOUT,FileName,len+1);

/* Теперь посылаем сообщение типа
MsgNewLine, которое содержит номер
строки и колонки */
MsgType=MsgNewLine;
WRITE(MsgType);

/* Определим номер строки с ошибкой */
len= (int) (PtrToErrorWord - 2
/* Указатель на конец номера строки */
- (AuxPtr+1)+1);
/* Указатель на начало номера строки */

memcpy(LineNumberString,AuxPtr+1,len);
/* Копируем номер строки */
LineNumberString[len]=0; /* Конец строки */
/* Переводим номер строки в число */
LineNumberWord=atoi(LineNumberString);

WRITE(LineNumberWord);
/* Выводим в stdout номер строки */

/* Номер колонки примем равным 1 */
ColumnNo=1;
WRITE(ColumnNo);

/* Теперь выводим сам текст сообщения об ошибке */
PtrToErrorWord++;
write(STDOUT,PtrToErrorWord,strlen(PtrToErrorWord)+1);
}
}

End:
MsgType=MsgEoFile;
WRITE(MsgType);
/* Записать в stdout признак окончания
работы с окном сообщений */
}

```



Фирма «Мир ПК» в рамках единой программы корпорации International Data Group (IDG) по изучению рынков вычислительной техники проводит адаптацию оригинальных методик исследований для рынков России и стран ближнего зарубежья. Мы предлагаем нашим читателям настоящую анкету. Ваши ответы помогут нам выявить наиболее актуальные на сегодняшний день направления исследовательской работы. Результаты этого опроса будут использованы как статистический материал для адаптации методик IDG по исследованию рынка вычислительной техники. Кроме того, мы рассматриваем этот опрос как первый шаг установления регулярной оперативной обратной связи с читательской аудиторией.

Ф.И.О. _____
 Фирма _____
 Должность _____
 Адрес _____

 страна индекс

 нас. пункт область

 ул. дом, корп., кв.

1. Деятельность в какой сфере компьютерного рынка Вас интересует

- ☐ Аппаратные средства
☐ разработка
☐ продажа
☐ Программное обеспечение
☐ разработка
☐ продажа
☐ зарубежное ПО
☐ отечественное ПО
☐ ЛВС
☐ Телекоммуникации
☐ Сервисные услуги
☐ Другая
☐ Коммерческие услуги

2. Какая информация вам необходима

- ☐ Технические описания
☐ Сравнительный анализ продуктов и услуг (каких) _____
 по каким критериям _____
☐ Экономические обзоры
☐ Практическое применение
☐ Прогноз тенденций по критериям _____
☐ Другое _____
 указать

Благодарим Вас за помощь в данной работе. Первым десяти читателям, приславшим заполненные анкеты, в качестве приза будет оформлена бесплатная подписка на оригинальное электронное издание «Мир ПК-диск» на 1993 г. *

* Дата отправки определяется по почтовому штемпелю.

Borland Pascal 7.0:

новые возможности

В.В. Фаронов

Описываются пакеты Borland Pascal 7.0 и Turbo Pascal 7.0, о выпуске которых на рынок фирма Borland объявила в ноябре 1992 г.

Вряд ли сегодня в бывшем СССР найдется хотя бы один программист, работающий на IBM-совместимых персональных компьютерах, который не пользовался бы продукцией фирмы Borland или хотя бы не слышал о ней. Пакет Turbo Pascal был первым коммерческим продуктом фирмы и сразу же стал популярным, а ныне он — мировой лидер по объемам продаж в классе коммерческих компиляторов. С помощью Turbo Pascal разрабатываются сложные современные СУБД и системы бухгалтерского учета, игры и средства автоматизированного проектирования, обучающие программы и многое, многое другое; ему посвящено множество публикаций (только на прилавках московского Дома книги в один из осенних дней я лично насчитал девять(!) разного объема и качества книг по Turbo Pascal); он широко используется в преподавании, вытесняя Бейсик из школ и Фортран из вузов.

Поэтому вполне понятно то внимание, с которым отечественные программисты относятся ко всем новостям, поступающим из далекого калифорнийского города Скоттс-Вэлли, где расположена штаб-квартира Borland. В ноябре 1992 г. там было объявлено о выпуске на рынок очередной — седьмой — версии компилятора языка Паскаль.

Собственно говоря, выпущен не один пакет, а сразу два — Borland Pascal with Objects 7.0 и Turbo Pascal 7.0. Система программирования Borland Pascal 7.0 ориентирована на разработчика-профессионала и позволяет создавать программы для процессоров 80286 и более новых, работающих под управлением DOS в двух режимах — реальном и защищенном (DPMI — DOS Protected Mode Interface) — и под управлением Windows. Она содержит две равные по возможностям интегрированные среды разработки

(ИСП) — BPW.EXE для среды Windows и BP.EXE для DOS, каждая из которых обеспечивает создание программ для DOS, Windows и DPMI.

Что же касается Turbo Pascal 7.0, то он представляет собой «облегченный» и удешевленный вариант Borland Pascal 7.0 и позволяет создавать программы только для реального режима DOS. Во всем остальном компилятор, входящий в ИСП TURBO.EXE представляет те же возможности, что и BP.EXE/BPW.EXE.

Московское отделение фирмы Borland любезно предоставило мне возможность детально познакомиться с обоими пакетами.

Пользуясь случаем, я хочу поблагодарить представителей фирмы за эту возможность и поделиться с читателями своими впечатлениями.

ТРЕБОВАНИЯ К АППАРАТНЫМ СРЕДСТВАМ

Честно говоря, когда я запустил инсталляционную программу INSTALL.EXE, то испытал легкий шок: на экране появилось предупреждение, что на моем жестком диске должно быть не менее 30 Мбайт свободного пространства! Кроме того, программа интересовалась, в каком каталоге размещается ядро Windows. Поскольку Windows у меня развернута не была, а о том, чтобы освободить на диске 30 Мбайт, не могло быть и речи, процедуру инсталляции пришлось прервать.

Кончилась эта история, в общем, достаточно благополучно — я «все-навсего» заменил машину на более или менее приличную по нынешним меркам: процессор 80386 с тактовой частотой 20 МГц, ОЗУ объемом 4 Мбайт. Предвижу горестный вздох, вырвавшийся из вашей груди, уважаемый читатель. Увы, сегодня такие параметры еще нельзя считать типичными для компьютеров, распространенных в нашей стране.

В сущности, требования пакета Borland Pascal 7.0 к аппаратным средствам — те же, что у системы Windows, на которую он в первую очередь ориентирован: процессор 80286 или более поздний, ОЗУ объемом не менее 2 Мбайт и дисплей EGA или лучше. Turbo Pascal 7.0 «демократичнее»: он занимает на жестком диске 5 Мбайт и работает с любым процессором и любым дисплеем.

НОВЫЕ ВОЗМОЖНОСТИ ИСР

Функционально ИСР BP.EXE и BPW.EXE полностью идентичны, но среду BPW отличает элегантность, присущая всем с любовью сделанным программам под Windows (в фирме Borland говорят о «ювелирном» программировании, понимая под этим, в частности, тщательную — до мельчайших деталей — продуманность пользовательского интерфейса).

В главном меню появилась новая опция *Tools*, из которой можно обращаться к внешним инструментальным программам, а в меню *Options* — подменю *Target*, где можно указать, какие именно программы следует сделать доступными из опции *Tools*. Стандартная настройка *Tools* для BP.EXE и TURBO.EXE включает Turbo Debugger, Turbo Assembler, Turbo Profiler и Grep, а для BPW.EXE — Turbo Debugger, Resource Workshop, WinSight и Turbo Profiler.

В меню *Compile* появилась опция *Target* и восстановлена присутствовавшая в версиях 4.0—5.5 опция *Information*. Первая позволяет явно указать платформу (т.е. DOS, Windows или DPMI), на которую рассчитана вновь создаваемая программа, а вторая дает исчерпывающую информацию о минимальных ресурсах, требуемых для ее выполнения. Как уже говорилось, компиляторы BP/BPW позволяют создавать программы разного типа, но создать программу еще не значит выполнить ее.

Запуск из среды BP.EXE программы для Windows, разумеется, невозможен (BP.EXE запускается из DOS), а вот запустить из нее DPMI-программу можно: вначале будет загружен автоматически создаваемый «запускач» — короткая программа, осуществляющая все необходимые действия для перевода процессора в защищенный режим (если, разумеется, перед этим он работал в реальном режиме).

Встроенный отладчик работает только с программами для реального режима DOS, но предоставляемый с пакетом внешний отладчик Turbo Debugger 3.2 работает с любыми программами, в том числе с предназначенными для Windows и DPMI. Если при компиляции программы установить опцию *Options—>Debugger—>Standalone debugging*, то в конец EXE-файла будет автоматически помещена отладочная информация, и работа с Turbo Debugger станет практически неотличимой от работы со встроенным отладчиком в режиме DOS. Однако Turbo Debugger поддерживает только текстовый режим экрана, и я должен сознаться, что чисто внешне переход от работающей под Windows графической среды

BPW.EXE к текстовой среде Turbo Debugger в момент начала отладки производит неприятное впечатление.

Во всех трех ИСР реализована возможность автоматически выделять цветом (а в BPW, кроме того, и шрифтом) различные конструкции языка: например, зарезервированные слова (*begin*, *for*, *then* и т.п.) пишутся белыми буквами, комментарии — серыми, константы и идентификаторы — желтыми. Это делает программы более «читаемыми» и облегчает отладку.

С помощью <Alt>-<F10> или правой кнопки «мыши» можно в любой момент вызвать контекстно-зависимое локальное меню с основными возможностями, доступными из текущего состояния среды.

Ряд усовершенствований коснулся встроенной подсказки. Появилась отдельная опция, позволяющая вызвать список всех стандартных примеров. В описаниях объектов Turbo Vision имеется ссылка на отдельное окно, содержащее графическое представление положения объекта в общей иерархии объектов Turbo Vision. Имеется также ссылка на графическое представление иерархии наследования методов и полей. Разработчики учли проблемы, возникающие при локализации программных продуктов (т.е. при их русификации, если речь идет о нашей стране), и предусмотрели возможность простой замены стандартного Help-файла.

Некоторые непринципиальные изменения коснулись и других опций среды. Однако в целом можно констатировать, что среда разработки программ мало изменилась по сравнению с Turbo Pascal 6.0, хотя и стала несколько более удобной.

ЯЗЫК И СИСТЕМНАЯ БИБЛИОТЕКА

Вряд ли стоит говорить о том, что обычно именно расширения языка вызывают наибольший интерес при знакомстве с новой версией компилятора. Есть такие расширения и в версии 7.0.

Появилось новое зарезервированное слово *Inherited* (унаследованный), которое можно использовать вместо имени родительского объекта при вызове унаследованных методов. Например:

```
type
  TMyProgram = object(TApplication)
    Procedure HandleEvent(var Event: TEvent); virtual;
  end;
  Procedure TMyProgram.HandleEvent;
begin
```



```
Inherited HandleEvent(Event);
.....
end;
```

При объявлении объекта теперь можно наряду с зарезервированным словом **Private** использовать и слово **Public**, что позволяет объявлять скрытыми (или доступными) поля и методы объекта в произвольном порядке (в версии 6.0 слово **Private** должно было открывать единственную секцию скрытых полей и методов в конце объявления объекта). Например:

```
type
  THiddenWord = object
    Private
      A: word;           {Скрытое поле}
    Public
      Procedure SetWord(W: word); {Доступные методы}
      Function GetWord: word;
end;
```

Заметим, что слова **Private** и **Public** резервируются только в контексте объявления объекта.

Как известно, в стандартном Паскале (и в предыдущих версиях Turbo Pascal) формальные параметры подпрограмм могут передаваться либо по ссылке, либо по значению: в первом случае подпрограмма способна повлиять на значение переданной ей в качестве параметра переменной вызвавшей программы, во втором — нет. Фактически при передаче по значению параметры перед вызовом подпрограммы копируются в стек, что приводит к большим потерям времени и памяти, если таким способом передаются крупные структурированные переменные (массивы и записи). В новой версии появился еще один тип параметров — константные. Они передаются по ссылке, но изменение их внутри подпрограммы запрещено синтаксисом языка, что контролируется компилятором. Константные параметры описываются в заголовке подпрограммы с помощью зарезервированного слова **Const**:

```
Procedure P(const A: byte);
```

После такого объявления внутри подпрограммы невозможен, например, оператор

```
A := NewValue;
```

В новой версии в качестве параметров подпрограмм могут использоваться так называемые «открытые массивы», которые описываются в заголовке подпрограммы с помощью зарезервированного слова **Array**, но без указания границ.

В процедуре **ArrayPrint** в листинге 1 формальный параметр **Arr** описан как открытый массив: указан лишь тип его элементов, но не указаны границы. Это дает возможность обращаться к процедуре с фактическими параметрами разной длины. Заметим, что новая стандартная функция **High** в этом случае возвращает верхнее возможное значение индекса при обращении к массиву, в то время как нижнее значение индекса открытых массивов в подпрограмме всегда равно нулю (разумеется, в вызывающей программе нижняя граница индексов может отличаться от нуля). Передавать таким способом можно только одномерные массивы. Если бы, например, в программе был объявлен массив

```
var
  c: array [0..1, 0..1] of byte;
```

то обращение

```
ArrayPrint(c)
```

было бы расценено компилятором как синтаксическая ошибка.

В языке предусмотрен новый стандартный тип **OpenString** (открытая строка), позволяющий передавать подпрограммам строки переменной длины, даже если включена опция *Options—>Compiler—>Strict var-string*. Эта возможность появляется только в том случае, когда программа компилируется с активизированной опцией *Option—>Compiler—>Open parameters* среды или с директивой компилятора **{SP+}**.

Нетрудно заметить, что открытые массивы и

Листинг 1. Пример использования открытых массивов.

```
Procedure ArrayPrint(Arr: array of byte);

var
  Max, k: word;

begin
  Max := High(Arr);
  WriteLn('Массив содержит ', Max+1, ' элементов:');
  for k := 0 to Max do
    Write(Arr[k]:4);
  WriteLn
end;

const
  a: array [1..3] of byte = (1, 2, 3);
  b: array [-1..3] of byte = (0, 2, 4, 6, 8);

begin
  ArrayPrint(a);
  ArrayPrint(b)
end.
```


открытые строки — это очередной шаг к сближению Паскаля с языком Си. Другим таким шагом является языковая поддержка ASCIIZ-строк. Принятый в Паскале способ внутреннего представления строк, когда нулевой байт содержит длину строки, неудобен тем, что ограничивает максимальную длину строки 255 символами. Как известно, в Си используется другой способ внутреннего представления строк: в конце цепочки байтов, образующих строку, ставится ограничитель — символ #0. Такой формат (он называется ASCIIZ-форматом) не накладывает ограничений на длину строки. Поддержка ASCIIZ-строк впервые появилась в библиотеке Turbo Vision еще в предыдущей версии 6.0. В новой версии этот формат получил «официальную» поддержку за счет введения нового типа данных PChar и стандартного модуля Strings. Тип PChar формально соответствует такому объявлению:

```
type
  PChar = ^Char;
```

Но фактически компилятор рассматривает переменные этого типа как указатели на ASCIIZ-строки. Более того, переменные типа PChar можно индексировать как массивы символов (см. листинг 2).

Новая стандартная функция StrNew, использованная в примере, размещает в динамической памяти строку и возвращает указатель на нее. Однако вывод содержимого вновь созданной ASCIIZ-строки осуществляется поэлементно в цикле, так, как если бы переменная S была объявлена следующим образом:

```
var
  S: array [0..4] of char;
```

Хотя в строго типизированном Паскале такое вольное смешение типов в принципе недопусти-

Листинг 2. ASCIIZ-строки.

```
{ $X+ }      { Включить расширенный синтаксис! }

Uses Strings;

var
  S: PChar;
  k: byte;

begin
  S := StrNew('12345');
  for k := 0 to 4 do
    Write(S[k]);
  Writeln
end.
```

мо, расширенный синтаксис Turbo Pascal 7.0 (он включается директивой компилятора { \$X+ }) делает подобные конструкции вполне возможными.

Заметим, что в модуле Strings функция StrNew определена следующим образом:

```
Function StrNew(Str: PChar): PChar;
```

Расширенный синтаксис позволяет считать совместимым с типом PChar любой массив символов с нулевым начальным индексом и любую текстовую константу. Если же компилировать программу из листинга 2, отключив расширенный синтаксис, компилятор выдаст сообщение:

Error 26: Type mismatch

и установит курсор сразу за константой '12345' в первой строке исполняемой части программы.

Следует отметить, что пример из листинга 2 достаточно искусствен, так как стандартные процедуры ввода-вывода поддерживают ASCIIZ-формат, и текст программы мог бы быть таким:

```
{ $X+ }
Uses Strings;
var
  S: PChar;
begin
  S := StrNew('12345');
  Writeln(S)      { Вывод содержимого строки! }
end.
```

В язык введена возможность контроля совместимости типов при использовании оператора @ (получение адреса). Теперь компиляция программы

```
{ $T+ }      { Включить контроль типов для оператора @ }
var
  W: Word;
  P: ^Longint;
begin
  P := @W
end.
```

вызовет сообщение о несовместимости типов для оператора присваивания. Контроль совместимости управляется новой опцией среды Options—>Compiler—>Typed @ operator или директивой компилятора { \$T+ /- }. На мой взгляд, это очень полезное нововведение, которое позволит уменьшить количество ошибок при работе с переменными процедурных типов.

Еще одна полезная мелочь — контроль пере-

полнения при арифметических операциях над переменными целочисленных типов: установив опцию *Option—>Compiler—>Overflow checking* или директиву компилятора `{ $Q+ }`, можно включить режим автоматической генерации кода, осуществляющего соответствующую проверку для операций сложения, вычитания и умножения, а также для стандартных функций `Abs`, `Sqr`, `Succ` и `Pred` (обращение к «быстрым» процедурам `Inc` и `Dec` по-прежнему не контролируется). Замечу, что директива `{ $R+ }` не дает нужного результата, так как в этом случае проверяются лишь границы типа-диапазона.

Строго говоря, описанная возможность не является языковой — ведь на уровне исходного текста нельзя контролировать динамическое изменение переменных. Проверка переполнения относится к усовершенствованиям компилятора, который теперь автоматически определяет

также наличие 32-разрядного процессора 80386/80486 и, если таковой обнаружен, использует его возможности для ускорения операций над длинными целыми числами.

В системную библиотеку `SYSTEM.TPU` добавлен ряд полезных процедур и функций.

В анонсе пакета утверждается, что улучшен доступ к элементам множеств. Сначала мне этого обнаружить не удалось: тестовая программа (см. листинг 3), откомпилированная с помощью Turbo Pascal 7.0, работала приблизительно на 14% медленнее, чем откомпилированная в среде Turbo Pascal 6.0.

Однако ситуация резко изменилась, когда вместо традиционных для Паскаля операций изменения множеств «+» и «-» я использовал две новые процедуры: `Include(var S: set of T; Item: T)` — включение элемента `Item` в множество `S` — и `Exclude` — исключение элемента из множества. После замены в тестовой программе операторов

```
S := S+[255];
S := S-[255]
```

на

```
Include(S,255);
Exclude(S,255)
```

скорость выполнения цикла увеличилась в 65 раз!

Приблизительно на 25% по сравнению с версией 6.0 и более чем на 30% по сравнению с 5.5 увеличилась скорость ввода-вывода текстовых файлов (измерения проводились с помощью программы, приводимой в листинге 4).

Две новые процедуры, включенные в библио-

Листинг 3. Тестирование скорости доступа к элементам множеств.

```
var
  OldTime: LongInt;

{В этой программе для получения системного времени
используется прямое обращение к области данных BIOS, где по адресу [$0000:$046C] располагается
четырёхбайтовый счетчик, обновляющийся приблизительно
каждые 55 миллисекунд.}
```

```
Procedure SetT;
{Запоминает текущее время в момент обновления
системного счетчика}

begin
  OldTime := MemL[$0000:$046C];
  while OldTime=MemL[$0000:$046C] do;
    OldTime := MemL[$0000:$046C]
  end;
```

```
Function GetT: LongInt;
{Возвращает разницу в «тиках» (по 55 мс) между текущим
временем и сохраненным в OldTime}

begin
  GetT := MemL[$0000:$046C]-OldTime
end;
```

```
var
  S: set of Byte;
  K: Word;
```

```
begin
  SetT; {Запоминаем текущее время}
  for K := 0 to 2*MaxInt+1 do
    begin
      S := S+[255]; {Помещаем элемент в множество}
      S := S-[255] {Удаляем элемент}
    end;
  Writeln(GetT*55/1000:6:1)
end.
```

Листинг 4. Тестирование скорости ввода-вывода текстовых файлов.

```
var
  OldTime: longInt;
```

```
Procedure SetT;

begin
  OldTime := MemL[$0040:$006C];
  While OldTime=MemL[$0000:$046C] do;
    OldTime := MemL[$0000:$046C]
  end;
```

```
Function GetT: LongInt;

begin
  GetT := MemL[$0040:$006C]-OldTime
end;
```



```

var
  F: text;
  S: string;
  K: Integer;

begin
  SetT;
  Assign(F, 'TEST.TXT');
  Rewrite(F);
  S := '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ';
  for K := 1 to 10000 do
    WriteLn(F, S);
  Close(F);
  Reset(F);
  While not EOF(F) do
    ReadLn(F, S);
  Close(F);
  Erase(F);
  WriteLn(GetT*55/1000:5:1);
end.

```

теку SYSTEM.TPU, могут оказаться полезными при организации циклов FOR, REPEAT и UNTIL:

Break — немедленный выход из цикла; эта процедура аналогична безусловному переходу (GOTO) на оператор, стоящий непосредственно после конца цикла;

Continue — досрочное завершение очередного прохода цикла — эквивалент перехода в конец цикла.

Пример использования процедуры Break приведен в листинге 5 (программа определяет, каким по счету будет целое число 100 в последовательности случайных чисел, генерируемой функцией Random).

В библиотеку введены две новые функции: High(X) возвращает верхнее, а Low(X) — нижнее значение типа-диапазона, к которому принадлежит целочисленная переменная X. Для параметра, переданного подпрограмме в виде открытого массива, функция High возвращает верхнее значение индекса, а для открытой строки — текущую длину строки.

В целом описанные выше изменения, хотя

Листинг 5. Использование процедуры Break.

```

var
  K: Integer;

begin
  K := 1;
  repeat
    If Random(255)=100 then
      Break;
    Inc(K);
  until K=MaxInt;
  WriteLn(K);
end.

```

они не затрагивают основ Паскаля и не будут, судя по всему, играть такую же значительную роль, как модули версии 4.0 или объекты версии 5.5, делают язык более гибким и увеличивают быстроедействие программ.

БИБЛИОТЕКА TURBO VISION

В новой версии в объектно-ориентированную библиотеку Turbo Vision добавлены:

- объект TValidator и его потомки, обеспечивающие контроль данных, получаемых от пользователя в ходе диалога;
- объект TOutLine, предоставляющий гибкие средства просмотра иерархических списков строк;

- метод TProgram.ExecuteDialog для вывода диалоговых окон и реализации диалога;

- возможность динамического запрета/разрешения выбора кнопок в кластерах (подобно тому, как запрещаются/разрешаются команды).

Абстрактный объект TValidator является производным от TObject и служит родоначальником иерархии родственных объектов, каждый из которых обычно ассоциирован с объектами типа TInputLine (строка ввода) и предназначен для контроля получаемых от пользователя текстовых данных. Практический интерес представляют собой четыре потомка этого объекта: TPXPictureValidator, TStringLookupValidator, TFilterValidator и TRangeValidator. Объект TPXPictureValidator проверяет введенную пользователем строку на соответствие эталонному образцу ввода — текстовой строке, подготовленной в соответствии со спецификациями СУБД Paradox. В объекте TStringLookupValidator введенная пользователем строка должна в точности соответствовать одной из строк-образцов, содержащихся в предварительно заданном списке. С помощью объекта TFilterValidator осуществляется контроль единственного введенного пользователем символа, а с помощью TRangeValidator строка ввода преобразуется в целое число и проверяется соответствие этого числа заданному диапазону.

Объекты TXXXValidator избавляют программиста от рутинной работы, связанной с проверкой наиболее популярных форматов ввода текстовых данных. Например, в программе, приведенной в листинге 6, создается диалоговое окно с двумя полями ввода: в первом поле требуется ввести целое число в диапазоне 1..2, во втором — дату в формате ДД-ММ-ГГ. При попытке закрыть окно нажатием клавиши <Enter>, про-

Листинг 6. Использование объектов контроля ввода.

```

Uses Objects,Validate,Dialogs,App,Views;

type
  TMyApp = object(TApplication)
    Procedure Run; virtual;
  end;

Procedure TMyApp.Run;

var
  R: TRect;
  W: PDialog;
  I,J: PinputLine;

begin
  R.Assign(10,5,70,14);
  W := New(PDialog,Init(R,""));
  with W ^ do
    begin
      R.Assign(10,3,40,4);
      I := New(PinputLine,Init(R,30));
      I ^ .SetValidator(New(PRangeValidator,Init(1,2)));
      Insert(I);

      R.Assign(5,2,30,3);
      Insert(New(PLabel,Init(R,'~ 0 ~ Целое 1..2:',I)));

      R.Assign(10,5,40,6);
      J := New(PinputLine,Init(R,30));
      J ^ .SetValidator(New(PPXPictureValidator,
        Init('{#[#]}-#[#]}-#[#]':True)));
      Insert(J);

      R.Assign(5,4,30,5);
      Insert(New(PLabel,Init(R,'~ 1 ~ Дата ДД-ММ-ГГ:',J)));

      R.Assign(45,2,55,4);

      Insert(New(PButton,Init(R,'~ O ~ kay',cmOk,bfDefault)));
    end;

    DeskTop ^ .ExecView(W)
  end;

var
  MyApp: TMyApp;

begin
  MyApp.Init;
  MyApp.Run;
  MyApp.Done
end.
    
```

грамма проверит ввод данных в полях. Если ввод был неверным, появится диагностическое сообщение и в диалоговом окне будет активизировано соответствующее поле ввода.

Типичным применением объекта TOutline является просмотр дерева каталогов на диске. Этот объект обеспечивает вывод на экран и прокрутку в вертикальном и горизонтальном направлениях древовидных структур следующего типа:

```

type
  PNode = ^TNode;
  TNode = record
    Next: PNode; {Ссылка на следующий узел
                  того же уровня}
    Text: PString; {Ссылка на текстовую строку}
    ChildList: PNode; {Ссылка на подчиненный список}
    Expanded: Boolean {Флаг видимости ChildList}
  end;
    
```

Поле Next указывает на следующий элемент списка данного уровня (для последнего элемента в списке это поле должно иметь значение NIL). Поле ChildList содержит ссылку на подсписок (список следующего уровня) или NIL, если такового нет. По умолчанию методы объекта создают на экране графическое представление иерархии элементов списка — примерно так же, как это делает Norton Commander по команде <Alt>-<F10> (показать дерево каталогов): слева от текстовой строки Text ^ выводятся символы псевдографики, наглядно показывающие место текущего элемента в полной иерархии списка. Предусмотрены методы для управления указателем и выбора элемента.

В объект TProgram введен новый метод ExecuteDialog, имеющий следующий заголовок:

```

Procedure ExecuteDialog(P: PDialog; Data: Pointer): Word;
    
```

Главное отличие этого метода от ExecView состоит в том, что в нем автоматически поддерживается связь с данными, получаемыми в ходе диалога: перед выводом окна P ^ в него помещаются начальные значения данных Data ^, а после закрытия окна из него считываются и помещаются в Data ^ новые значения. При этом данные контролируются методами TXXXValidator.

Поле Value в объектах TCluster теперь имеет 32 разряда, что позволяет включать в кластер до 32 независимых кнопок и практически неограниченное число зависимых. Кроме того, добавлено 32-разрядное поле EnableMask, биты которого маскируют выбор первых 32 кнопок: если бит равен 0, нажатие соответствующей кнопки запрещено (в этом случае надпись выводится пониженной яркостью). Для воздействия на поле EnableMask служит метод SetButtonState.

ПОДДЕРЖКА ПРОГРАММИРОВАНИЯ ДЛЯ СРЕД WINDOWS И DPMI

До настоящего времени программисты, работавшие на Turbo Pascal, решив писать програм-

мы для Windows, должны были переходить либо на Си, либо на специальную версию Turbo Pascal for Windows, и в обоих случаях изучение нового языка и среды Windows требовало значительных усилий и времени.

Пакет Borland Pascal 7.0 делает этот переход значительно более плавным: как мы видели, изменения среды разработки, языка и стандартных библиотек не являются принципиальными и носят скорее количественный, чем качественный характер, так что пользователю Turbo Pascal 6.0 многое покажется знакомым.

Разработчики включили в пакет специальный модуль WinCRT, предельно упрощающий переход к новой среде. Подавляющее большинство ранее созданных вами программ, если только они использовали текстовый режим экрана, смогут работать под управлением Windows — для этого достаточно указать ссылку на WinCRT в предложении Uses и перекомпилировать программу с помощью BPW.EXE.

Имеющаяся в Borland Pascal 7.0 возможность создания программ для DOS в защищенном режиме уникальна: она не поддерживается в явной форме никакими другими системами программирования. Защищенный режим предоставляет две принципиально новые возможности по сравнению с реальным: непосредственное использование всей имеющейся в ПК оперативной памяти (до 16 Мбайт для процессоров 80286 и до 4 Гбайт для 80386/80486) и полную независимость друг от друга нескольких одновременно находящихся в памяти программ. Кроме того, Borland Pascal 7.0 позволяет в защищенном режиме создавать динамически загружаемые библиотеки (DLL) для DOS и работать с ними, по-

степенно готовя таким способом переход к Windows (формат DLL для DOS и Windows один и тот же).

Защищенный режим, реализуемый средствами Borland Pascal 7.0, соответствует промышленному стандарту DPMI 0.9 (DOS Protected Mode Interface) и позволяет создавать программы, которые почти не зависят от аппаратных особенностей конкретного компьютера.

Драйвер защищенного режима включает сервер DPMI (файл DPMI16BI.OVL), который обеспечивает поддержку режимов работы Windows 3.x, системного прерывания INT 21 MS-DOS и всех прерываний BIOS, и администратор DPMI (файл RTM.EXE), обеспечивающий управление таблицами дескрипторов, перевод микропроцессора в защищенный или реальный режим, распределение основной памяти DOS (первый мегабайт) и дополнительной памяти, управление обработкой прерываний и связью с программами, работающими в реальном режиме. Он совместим с Windows 3.x и с OS/2 1.x и может загружать программы из EXE- и DLL-файлов.

Администратор DPMI распределяет блоки памяти между обращающимися к нему прикладными программами. Он поддерживает фиксированные (fixed), перемещаемые (moveable) и выгружаемые (discardable) блоки (терминология Windows), а кроме того, способен автоматически загружать и выгружать оверлейные файлы, так что в этом смысле программы, работающие под его управлением, не нуждаются в модуле Overlay.

Доступ к администратору DPMI из программ обеспечивается модулем WinAPI. Этот модуль реализует наиболее важные функции интерфейса Windows для прикладных программ, в том

ЧТО ТАКОЕ ЗАЩИЩЕННЫЙ РЕЖИМ

Как известно, главное отличие защищенного режима от реального заключается в способе преобразования логических 32-разрядных адресов, используемых программой, в физические адреса, с которыми работает процессор. В реальном режиме логический адрес состоит из номера сегмента (старшие 16 бит) и смещения внутри сегмента (младшие 16 бит). Для получения физического адреса номер сегмента умножается на 16 (смещается влево на 4 разряда) и складывается со смещением. Такой способ

дает возможность адресовать лишь $(1024 + 64) \cdot 1024 = 1\,114\,096$ байт. Замечу, что старшие 64 Кбайт, выходящие за рамки адресного пространства размером 1 Мбайт (так называемая память HMA — High Memory Area), доступны при этом только в особом режиме, разрешающем использование 21-разрядной адресной шины. В защищенном режиме сегментная часть логического адреса представляет собой номер элемента так называемой таблицы дескрипторов. Каждый эле-

мент этой таблицы содержит базовый адрес (24-разрядный для процессора 80286 или 32-разрядный для процессора 80386/80486), к которому прибавляется смещение. Иными словами, в реальном режиме память адресуется сегментной частью адреса с точностью до параграфа (16 байт), а в защищенном — с точностью до сегмента. Помимо базового адреса в таблице дескрипторов хранится также длина сегмента (в защищенном режиме сегмент может иметь любую длину от 1 до 65536 байт)

и флаг доступа, определяющий, будут ли данные, находящиеся внутри сегмента, только читаться или они могут также записываться. Сегменты кода всегда доступны только для чтения, сегменты данных — для записи и чтения. Попытка выполнить код, записанный в сегменте данных, изменить байт в сегменте кода или выйти за пределы сегмента вызывает немаскируемое аппаратное прерывание, которое обрабатывается драйвером защищенного режима.

числе управление памятью, модулями, ресурсами и загрузку динамических библиотек.

Разработка программ для защищенного режима предельно проста. В большинстве случаев достаточно откомпилировать программу в расчете на этот режим работы и запустить ее из среды BP.EXE или BPW.EXE. Тем не менее специфика использования памяти в защищенном режиме накладывает на программы некоторые ограничения. В частности, при обращении к массивам Mem, MemW и MemL программист должен быть уверен в том, что базовый адрес дескриптора указывает на нужную часть памяти. Например, обращение к сегменту \$0040 будет соответствовать обращению к области данных BIOS только в том случае, если базовый адрес дескриптора равен \$0040, т.е. если программа размещена в основной памяти DOS. Чтобы сделать программу независимой от ее физического размещения в памяти, программисту необходимо использовать следующие глобальные переменные модуля System:

- Seg0040 — доступ к области данных BIOS;
- SegA000 — доступ к видеопамати адаптеров EGA и VGA в графическом режиме;
- SegB000 — доступ к видеопамати адаптера MDA;
- SegB800 — доступ к видеопамати адаптера CGA во всех режимах и EGA и VGA в текстовом режиме.

Например, вместо оператора

```
Time := MemL[$0040:$006C]
```

с помощью которого в переменную Time помещается текущее системное время, следует использовать оператор

```
Time := MemL[Seg0040:$006C]
```

Аналогично, нельзя использовать с зарезервированным словом Absolute ссылку на абсолютный адрес, если только сегментная часть адреса не содержит переменную SegXXXX. Очень осторожно следует применять сегментную арифметику. Например, если к сегменту добавить \$1000, то в защищенном режиме это вовсе не обязательно будет означать смещение в памяти на 64 Кбайт. Для перехода к смежному сегменту в защищенном режиме должна использоваться специальная функция без параметров SelectorInc.

Разумеется, необходимость в использовании этой функции возникает только в случае размещения в памяти крупных массивов данных (больше 64 Кбайт). Такое размещение возможно

с помощью функции GlobalAlloc. Крупные блоки соответствуют непрерывной последовательности дескрипторов, каждый из которых (кроме, возможно, последнего), ссылается на сегмент размером 64 Кбайт. Если, например, в программе использован вызов

```
Uses WinAPI;
var
  Buff: THandle;      {Ссылка на дескриптор}
  Size: LongInt;      {Длина блока}
begin
  Size := 300*1024;    {Резервировать 300 Кбайт}
  Buff := GlobalAlloc(gmem_Moveable, Size);
  .....
end.
```

то будут созданы пять дескрипторов, причем первые четыре будут ссылаться на сегменты длиной 64 Кбайт, а последний — на сегмент длиной $300 - 64 * 4 = 44$ Кбайт.

В листинге 7 приводится пример DPMI-программы, считывающей в память дисковый файл большого размера. Обратите внимание на использование функции SelectorInc: для перемещения «вверх» по памяти (от начала логического адресного пространства к его концу) достаточно нужное число раз прибавить возвращаемое ею значение к начальному указателю. Эта функция определена и для реального режима, где она возвращает значение \$1000 (соответствует смещению в физическом адресном пространстве на 64 Кбайт).

Листинг 7. Работа с памятью в защищенном режиме.

```
Uses WinAPI;

Function GetPtr(Selector: THandle;
  Offset: LongInt): Pointer;
{Возвращает указатель на область памяти внутри
крупного блока}
type
  Long = record
    Lo, Hi: Word;
  end;
begin
  GetPtr := Ptr(Selector +
    Long(Offset).Hi*SelectorInc,
    Long(Offset).Lo)
end;

Function LoadFile
  (const FileName: string): THandle;
{Размещает в памяти файл и возвращает ссылку на
него}
var
  Buffer: THandle;
```



```

Size,      {Длина блока}
Offset,    {Смещение в блоке}
Count: LongInt; {Остаток файла}
F: file;
begin
  Buffer := 0;
  Assign(F, FileName);
  {$I-}
  Reset(F,1);
  {$I+}
  if IOResult=0 then
  begin
    Size := FileSize(F);
    Buffer := GlobalAlloc(gmem_Moveable,Size);
    if Buffer < > 0 then
    begin
      Offset := 0;
      while Offset < Size do
      begin
        Count := Size-Offset;
        if Count > $8000 then
          Count := $8000;
        BlockRead(F,
          GetPtr(Buffer,Offset) ^,
          Count);
        Inc(Offset,Count)
      end
    end
  end;
  LoadFile := Buffer
end;

```

При обращении к функции GlobalAlloc требуется указать не только размер блока в байтах, но также и флаг перемещения блока. В качестве флага может использоваться одна из следующих констант, определенных в WinAPI:

```

gmem_Fixed      {Блок не может перемещаться}
gmem_Moveable   {Блок может перемещаться}
gmem_Discardable {Блок может выгружаться}

```

В модуле WinAPI определены более 20 разного рода подпрограмм, облегчающих доступ к дополнительной памяти, а также специальные процедуры и функции для доступа к администратору модулей и администратору ресурсов. Эти части драйвера защищенного режима осуществляют доступ соответственно к динамически загружаемым библиотекам и файлам ресурсов.

ОБ АВТОРЕ

Валерий Васильевич Фаронов — преподаватель Учебно-инженерного центра МВТУ — ФЕСТО Дидактик, автор книг «Программирование на персональных ЭВМ в среде Турбо-Паскаль» (М.: Издательство МГТУ, 1990) и «Турбо Паскаль» (в 3-х книгах). Кн. 1. «Основы Турбо Паскаля» (М.: МВТУ — ФЕСТО Дидактик, 1992). Контактные телефоны (095)263-60-98 и (095)267-49-84.

НОВОСТИ

Учебный центр фирмы SUN Microsystems

15 декабря 1992 г. состоялось открытие совместной лаборатории фирмы SUN Microsystems и Московского университета «Системы SUN в образовании и научных исследованиях».

В рамках этого проекта фирма SUN Microsystems, занимающая ведущее место в мире в области производства рабочих станций, безвозмездно передала Московскому университету несколько мощных рабочих станций и комплект ПО, а также взяла на себя обязательство регулярно обновлять все оборудование.

Лаборатории присваивается статус авторизованного учебного центра фирмы SUN (первого на территории Восточной Европы и бывшего Советского Союза). Перед лабораторией стоят следующие задачи:

- создание учебных компьютерных курсов и разработка программного обеспечения для обучения (с использованием технологии мультимедиа, совмещающей трехмерную графику, звук и производительность рабочих станций SUN);
- ведение научно-исследовательской работы сотрудниками и студентами на базе новейшего оборудования (прежде всего рабочих станций и серверов) и ОС UNIX;
- разработка технологии переноса отечественного программного обеспечения на современные мощные компьютерные платформы.

Лаборатория планирует принять активное участие в объединении вычислительных ресурсов МГУ в единую сеть, а также проводить консультации и давать экспертные оценки различным проектам в области информационных технологий.

Новые пакеты Symantec

Москва. 8 декабря 1992 г. поступила в продажу русскоязычная версия интегрированного пакета Q&A 4.0. Пакет включает в себя средства разработки и ведения баз данных, составления отчетов, набор утилит. Q&A имеет также мощный текстовый процессор.

Примечательно то, что обе программы фирмы Symantec — и выпущенная ранее Time Line (см. стр. 89), и Q&A — первые программные продукты, произведенные на Казанском комбинате программных средств. По оценке фирмы, качество производства — очень хорошее.

Купертино (шт. Калифорния). 7 декабря 1992 г. Symantec объявила о начале поставок своих новых пакетов программ: Norton Commander for OS/2 (первой сервисной программы, выпущенной для OS/2 независимой компанией) и The Norton Essentials 1.0. для ПК-блокнотов PowerBook фирмы Apple. Пакет The Norton Essentials 1.0 увеличивает время работы батарей до 4 часов и более, автоматически синхронизирует обмен данными между PowerBook и другими системами Macintosh, а также предлагает пользователю обширный набор сервисных программ.

Михаил Титкин

Новогодний подарок Microsoft

Корпорация Microsoft пообещала бесплатно предоставить легальным пользователям СУБД FoxPro 2.0 (198\$) расширение (upgrade) этой СУБД до готовящейся к выпуску версии 2.5. Новая Microsoft FoxPro будет работать под DOS, Windows, Unix и Macintosh.

Как поставить точку

В.Г. Чертков

На примере рисования точки рассматривается работа с видеопамятью в основных графических режимах адаптеров EGA и VGA.

Вряд ли нужно объяснять программисту, имеющему дело с графикой, что при «прямом» (почему кавычки, будет сказано чуть позже) доступе к видеопамати скорость рисования графических примитивов в несколько раз выше, чем при использовании функций BIOS. И уж совершенно очевидно, что любой графический объект представляет собой совокупность точек. Исходя из этих двух положений, мы на примере рисования точки рассмотрим организацию видеопамати и способы программирования видеоадаптеров EGA и VGA в основных графических режимах (разрешение 640×350 и 640×480 пикселей соответственно, объем видеопамати не менее 256 Кбайт).

Сразу же выделим три главные особенности видеопамати (ВП). Во-первых, ВП разделена на четыре блока, называемых параллельными битовыми плоскостями (или битовыми матрицами). Слово «параллельный» означает, что каждый адрес в видеопамати относится не к одному байту, а к четырем — по одному в каждой битовой плоскости.

Во-вторых, вследствие этого разделения доступ к ячейкам ВП осуществляется не напрямую, как в обычном ОЗУ, а через 8-разрядные шлюзы, каждый из которых связан с одной битовой плоскостью (отсюда и кавычки при слове «прямой»). И, наконец, в-третьих, процессы чтения и записи в ВП управляются контроллерами видеоадаптера, программирование которых предоставляет широчайшие возможности для создания графических объектов и управления ими. Схема организации видеопамати представлена на рис. 1.

ВИДЕОПАМЯТЬ И ЭКРАН МОНИТОРА

Пикселу на экране монитора соответствует по одному биту в каждой битовой плоскости. По сути, экран представляет собой построчную развертку одной битовой плоскости, причем битовые плоскости ВП как бы наложены одна на другую. В результате на один пиксел приходится четыре бита в ВП (рис. 1), а ячейки ВП отображаются на экране построчно, с увеличением адреса слева направо (вдоль каждой строки) и сверху вниз (см. рис. 2).

Так, пиксел с координатами (0,0) соответствует старшему биту в байте, имеющему смещение 0 относительно начала ВП, а пиксел с координатами (7,0) — младшему биту этого же байта. Байт ВП со смещением 1 «охватывает» пикселы с координатами от (7,0) до (15,0) и т.д. до конца строки, после него отображение ВП продолжается с начала следующей строки.

ЦВЕТ И ПАЛИТРА

Итак, на один пиксел приходится четыре бита ВП, и именно поэтому в основных графических режимах на EGA и VGA можно одновременно

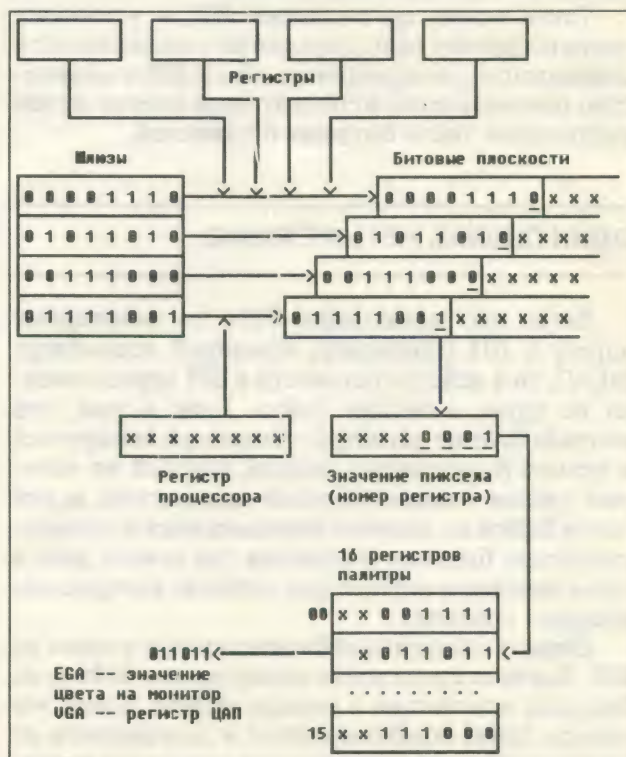


Рис. 1. Общая схема управления видеопаматью адаптеров EGA и VGA.

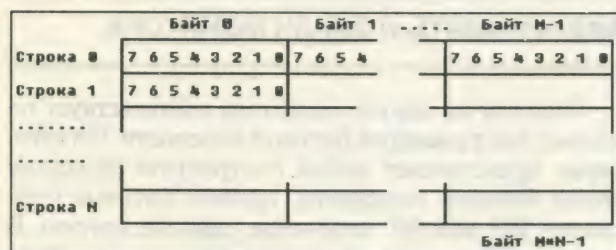


Рис. 2. Отображение видеопамати на экране монитора.

пользоваться 16 цветами ($2^4=16$). Всего же доступны 64 различных цвета, поскольку значение пиксела, записанное в четырех битах — не значение самого цвета, а номер одного из 16 регистров палитры, в которых записаны шестибитовые ($2^6=64$) значения цветов (см. рис. 1).

При использовании, например, процедуры SetAllPalette (Turbo Pascal) вы записываете в 16 шестибитовых регистров палитры 16 (из 64 возможных) значений цветов. Процедура же SetColor на самом деле устанавливает не цвет, а один из 16 регистров с записанным в нем значением цвета (это справедливо только для EGA; на VGA в регистре палитры хранится номер одного из регистров цифро-аналогового преобразователя, и лишь в нем содержится само значение цвета).

Такая схема организации ВП и установки цвета позволяет при сохранении совместимости видеоадаптеров наращивать объем ВП и количество одновременно используемых цветов путем увеличения числа битовых плоскостей.

ОДИН ПИШЕМ, ЧЕТЫРЕ В УМЕ

Когда вы записываете байт по некоторому адресу в ВП (например, командой ассемблера MOV), то в действительности в ВП пересылаются не один, а четыре байта. Дело в том, что вначале байт из регистра процессора копируется в четыре 8-разрядных шлюза, каждый из которых связан с одной битовой плоскостью, и уже затем байты из шлюзов пересылаются в соответствующие битовые плоскости (на самом деле в этом описании опущен ряд деталей, которые мы обсудим позднее).

Обратная картина наблюдается при чтении из ВП. Вначале происходит копирование байтов из битовых плоскостей в четыре шлюза. Далее эти четыре байта комбинируются в зависимости от содержимого многочисленных регистров видеоадаптера и пересылаются (что, впрочем, совсем не обязательно) в регистр процессора.

При такой схеме чтения следует с осторожностью пользоваться пересылкой слов. Посмотрите, что произойдет при замене, например, команды ассемблера LODSB (чтение байта) на команду LODSW (чтение слова). В первом случае однобайтовые «сторожевые катера» из океана видеопамати полностью входят в свои 8-разрядные шлюзы, и далее их комбинация поступает в регистр процессора. При использовании же команды LODSW вначале выполняются два последовательных чтения байта, а затем две записи. По этой причине двухбайтовый «авианосец» потребует «распилить» на две части и шлюзовать по очереди, причем «нос» придется прямо в шлюзе «затопить», иначе не войдет «корма». В результате при пересылке из шлюзов в регистр процессора вы получите только две «кормы».

РЕГИСТРЫ И КОНТРОЛЛЕРЫ

Многие программисты-любители (и не только они) не используют возможности «прямого» доступа к ВП главным образом из-за обилия регистров, которые необходимо программировать до и в процессе рисования графических изображений. Чтобы вы имели приблизительное представление о функциях видеоадаптера, перечислим его контроллеры, указав для каждого число регистров:

- блок синхронизации — 6 регистров;
- контроллер электронно-лучевой трубки — 28 регистров;
- графический контроллер — 12 регистров;
- контроллер атрибутов — 7 регистров;
- цифро-аналоговый преобразователь (только VGA) — 7 регистров.

Добавив к этому списку еще пять внешних регистров, получим в общей сложности 62 регистра (детальную информацию по видеоадаптерам вы можете найти в [1—3]). Но не торопитесь закрывать журнал — во-первых, для построения основных графических примитивов используются в основном не более пяти-шести регистров, а во-вторых, прямое программирование видеоадаптера предоставляет несравненно более богатые возможности, чем стандартные графические библиотеки таких пакетов, как Turbo Pascal или Turbo C.

Как уже упоминалось, при записи и чтении ВП происходит не просто пересылка байтов через шлюзы: возможны также различные опера-

Таблица 1. Регистры графического контроллера.

Порт	Индекс	Имя	Значение по умолчанию
3CEh	—	Индексный регистр (ИР)	00h
3CFh	0	Установка/сброс (УС)	00h
3CFh	1	Разрешение установки/сброса (РУС)	00h
3CFh	2	Сравнение цветов (СЦ)	00h
3CFh	3	Циклический сдвиг данных (ЦСД)	00h
3CFh	4	Выбор читаемой плоскости (ВЧП)	00h
3CFh	5	Режим чтения/записи (РЧЗ)	00h
3CFh	6	Смешанный (*)	05h
3CFh	7	Цвет безразличен (ЦБ)	0Fh
3CFh	8	Битовая маска (БМ)	FFh

* В настоящей статье не рассматривается.

ции над пересылаемыми данными в зависимости от содержимого регистров видеоадаптера. При работе в основных графических режимах на видеоадаптерах EGA и VGA используются главным образом регистры графического контроллера (ГК), перечисленные в табл. 1 [1].

Доступ ко всем регистрам ГК (как и к другим регистрам) осуществляется косвенно — через регистр ИР: в порт 3CEh записывается индекс требуемого регистра, а в порт 3CFh — значение, которое заносится в этот регистр. Например, запись значения FFh в регистр БМ происходит так (здесь и далее используется встроенный ассемблер пакета Turbo Pascal 6.0):

```
MOV DX,3CEh
MOV AL,8
OUT DX,AL      { записали в регистр ИР }
               { индекс регистра БМ }

MOV DX,3CFh
MOV AL,0FFh
OUT DX,AL      { записали в регистр БМ }
               { значение FFh }
```

На практике применяется более короткая последовательность команд: в регистр AL записывается индекс регистра, в AH — значение, а операнды команды OUT представляются в формате слов:

```
MOV DX,3CEh
MOV AL,8
MOV AH,0FFh
OUT DX,AX
```

На языке Turbo Pascal, который используется здесь в примерах наряду с ассемблером, запись становится еще короче:

```
PortW[$3CE]:=$FF08;
```

Кроме регистров ГК довольно часто используется один из регистров блока синхронизации.

Это регистр маскирования плоскости (МП), четыре младших бита которого запрещают или разрешают доступ к четырем битовым плоскостям (по одному биту на плоскость) во время записи. Доступ к регистру осуществляется путем записи его индекса (2) в порт 3C4h и значения в порт 3C5h. Значение регистра по умолчанию — 0Fh.

Программируя перечисленные регистры, можно управлять пересылаемыми через шлюзы значениями как бы в двух измерениях. Одни операции являются побайтовыми, т.е. влияют на содержимое каждого из шлюзов отдельно, другие — попиксельными, влияющими на один и тот же бит во всех четырех шлюзах.

Вы можете делать практически все, что вашей душе угодно: копировать содержимое шлюзов в ВП и обратно; пересылать содержимое любого из шлюзов в регистр процессора; комбинировать четырехбитовое значение из регистра процессора с любым или всеми значениями пикселей в шлюзах; использовать восьмибитовое значение из регистра процессора в качестве маски, указывающей, какой из восьми зашлюзованных пикселей пересылается в ВП и т.д.

РЕЖИМЫ ЧТЕНИЯ И ЗАПИСИ

Побайтовые и попиксельные операции в заранее определенной последовательности используются видеоадаптером в запрограммированных режимах записи и чтения. В EGA предусмотрены два режима чтения (0 и 1) и три режима записи (0, 1 и 2), а VGA в дополнение к ним имеет еще один режим чтения. Режимы устанавливаются путем записи соответствующего значения в регистр РЧЗ: в биты 0 и 1 записывается номер режима записи, в бит 3 — номер режима чтения. По умолчанию BIOS устанавливает режим чтения 0 и режим записи 0.

Почти для всех режимов чтения и записи ниже приведены примеры считывания/рисования пиксела, снабженные комментариями, которые помогут вам разобраться в том, как функционирует тот или иной режим. Процедуры и функции написаны на ассемблере и предназначены для использования в среде Turbo Pascal 6.0 (структура некоторых процедур заимствована из [2]). Для тех, кто не знаком с ассемблером или не имеет версии 6.0, эти же процедуры приведены также на языке Turbo Pascal. Кроме того, в отличие от ассемблерных процедур, более детально раскрывающих процессы чтения и записи, процедуры на Паскале компактнее и благодаря этому легче читаются.

Теперь запаситесь терпением и сосредоточьтесь.

ЧТЕНИЕ/ЗАПИСЬ
ВИДЕОПАМЯТИ

Режим чтения 0 (листинг 1). Вначале происходит загрузка шлюзов из соответствующих

ших битовых плоскостей, после чего содержимое одного из шлюзов пересылается в регистр процессора (см. рис. 3). Выбор шлюза осуществляется путем предварительной записи номера плоскости (0—3) в регистр ВЧП. Этот регистр используется только в данном режиме.

Листинг 1. Чтение значения пиксела в режиме 0.

```
{*****}
Function GetPixelAsm0
  (X,Y : Integer): word; assembler;
{ Режим чтения 0 }
{ используется регистр ВЧП }
{ значение пиксела формируется по 1 биту; }
{ биты читаются из битовых плоскостей поочередно }
asm
  { вычисление битовой маски и адреса пиксела }
  MOV AX,Y    { AX = Y }
  MOV BX,X    { BX = X }
  MOV CH,BL   { CH - младший байт от X }
  MOV DX,80
  MUL DX     { AX = Y*80 (байт на строку) }

  MOV CL,3
  SHR BX,CL   { BX = X/8 }
  ADD BX,AX   { BX = байтовое смещение в ВП }

  MOV SI,BX
  MOV AX,0A000h { AX = сегментный адрес ВП }
  MOV ES,AX   { ES:SI = байтовый адрес }
  { пиксела }

  MOV CL,CH   { CL = младший байт от X }
  AND CL,7
  XOR CL,7    { битовая маска должна быть }
  { сдвинута на CL разрядов }
  { влево }
  MOV BL,1    { BL = исходная битовая маска }

  SHL BL,CL   { BL = битовая маска в }
  { заданной позиции }
  MOV BH,8    { BH = бит в пикселе, }
  { соответствующий }
  { читаемой битовой плоскости }
  XOR CH,CH   { обнуляем накопитель }
  { значения пиксела }

  { установка регистра ВЧП }
  MOV DX,3CEh { DX = порт регистра ИР }
  MOV AX,0304h { AH = начинаем чтение с }
  @ M1: OUT DX, AX { третьей битовой плоскости }
  { AL = номер регистра }
  TEST BL,ES:[SI] { проверяем, установлен ли }
  { бит в пикселе для текущей }
  { битовой плоскости }

  JZ @M2
  OR CH,BH    { если установлен, заносим }
  { бит в текущую позицию }
  { накопителя значения пиксела }
  @ M2: SHR BH, 1 { перемещаем бит в следующую }
  { позицию }
  DEC AH     { выбираем следующую битовую }
  { плоскость }
```

```
JGE @M1 { цикл по всем битовым плоскостям }

MOV AL,CH { значение функции }
XOR AH,AH { возвращается в регистре AX }

end; { GetPixelAsm0 }

{*****}
Function GetPixelPas0(X,Y : Integer): word;
{ Режим чтения 0 }
{ используется регистр ВЧП }
{ значение пиксела формируется по 1 биту; }
{ биты читаются из битовых плоскостей поочередно }

Var I : Integer;
R : registers;
ByteOfs : word;

begin
  ByteOfs:=80 * Y + X shr 3;
  { байтовое смещение в ВП }

  R.AH:= 1 shl ((X and 7) xor 7);
  { AH = битовая маска }

  R.CL:=1; { CL = бит в пикселе, }
  { соответствующий читаемой }
  { битовой плоскости }
  R.CH:=0; { обнуляем накопитель }
  { значения пиксела }
  R.BL:=4; { BL = индекс регистра ВЧП }

  { начинаем чтение с битовой плоскости 0 }
  For I:=0 to 3 do
  begin
    R.BH:=I; { BH = номер текущей }
    { битовой плоскости }
    PortW[$3CE]:=R.BX;
    { установка регистра ВЧП }

    { Проверяем, установлен ли бит в пикселе }
    { для текущей битовой плоскости. Если }
    { установлен, заносим бит в текущую }
    { позицию накопителя значения пиксела }
    If Mem[$A000:ByteOfs] and R.AH <> 0 then
      R.CH:=R.CH or R.CL;

    R.CL:=R.CL shl 1; { перемещаем бит в }
    { следующую позицию }
  end;

  GetPixelPas0:=R.CH;
  { GetPixelPas0 = значение пиксела }

  PortW[$3CE]:=$0004;
  { восстанавливаем значение регистра }
  { ВЧП по умолчанию }

end; { GetPixelPas0 }

{*****}
```

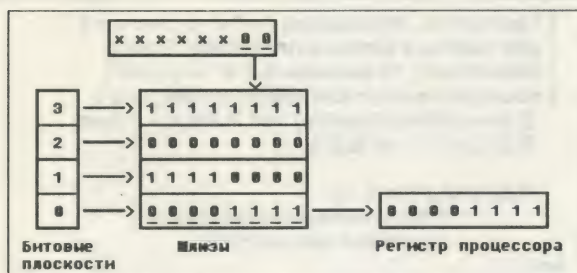



Рис. 3. Режим чтения 0.

В режиме чтения 1 (листинг 2) также вначале выполняется пересылка байтов из битовых плоскостей в шлюзы. Затем все восемь зашлюзованных пикселей сравниваются со значением, записанным в регистре СЦ. Результат сравнения в виде одного байта данных пересылается в регистр процессора. Значение этого байта определяется так: при совпадении значения пикселя со значением в регистре СЦ бит в соответствующей пиксельной позиции устанавливается в 1, при несовпадении — в 0 (см. рис. 4).

Листинг 2. Чтение значения пикселя в режиме чтения 1.

```
{*****}
```

Function GetPixelAsm1

```
(X,Y : integer): word; assembler;
```

```
{ Режим чтения 1 }
```

```
{ используются регистры СЦ и ЦБ }
```

```
{ значение пикселя формируется по 1 биту; биты }
```

```
{ читаются из битовых плоскостей поочередно }
```

asm

```
{ вычисление битовой маски и адреса пикселя }
```

```
MOV AX,Y { AX = Y }
```

```
MOV BX,X { BX = X }
```

```
MOV CH,BL { CH - младший байт от X }
```

```
MOV DX,80
```

```
MUL DX { AX = Y*80 (байт на строку) }
```

```
MOV CL,3
```

```
SHR BX,CL { BX = X/8 }
```

```
ADD BX,AX { BX = байтовое смещение в ВП }
```

```
MOV SI,BX
```

```
MOV AX,0A000h { AX = сегментный адрес ВП }
```

```
MOV ES,AX { ES:SI = байтовый адрес }
```

```
{ пикселя }
```

```
MOV CL,CH { CL = младший байт от X }
```

```
AND CL,7
```

```
XOR CL,7 { битовая маска должна быть }
```

```
{ сдвинута на CL разрядов влево }
```

```
MOV BL,1 { BL = исходная битовая маска }
```

```
SHL BL,CL { BL = битовая маска в заданной }
```

```
{ позиции }
```

```
XOR CH,CH { обнуляем накопитель значения }
```

```
{ пикселя }
```

```
{ установка регистра P43 }
```

```
MOV DX,3CEh { DX = порт регистра IP }
```

```
MOV AX,0805h
```

```
{ AH = режим записи 0 (биты 0,1) }
```

```
{ режим чтения 1 (бит 3) }
```

```
{ AL = индекс регистра }
```

```
OUT DX,AX
```

```
{ установка регистра СЦ }
```

```
MOV AX,0F02h
```

```
{ AH = 1111B: значение регистра }
```

```
{ AL = индекс регистра }
```

```
OUT DX,AX
```

```
{ установка регистра ЦБ }
```

```
MOV AX,0807h { AH = начинаем чтение с }
```

```
{ третьей битовой плоскости }
```

```
{ AL = индекс регистра }
```

```
@M1: OUT DX,AX
```

```
TEST BL,ES:[SI] { проверяем, установлен ли }
```

```
{ бит в пикселе для текущей }
```

```
{ битовой плоскости }
```

```
JZ @M2
```

```
OR CH,AH { если установлен, заносим }
```

```
{ бит в текущую позицию }
```

```
{ накопителя значения пикселя }
```

```
@ M2: SHR AH, 1 { перемещаем бит в следующую }
```

```
{ позицию, выбирая следующую }
```

```
{ битовую плоскость }
```

```
JNC @M1 { цикл по всем битовым плоскостям }
```

```
{ восстановление значения регистров по умолчанию }
```

```
MOV AX,0005h { регистр P43: режим записи 0, }
```

```
OUT DX,AX { режим чтения 0 }
```

```
MOV AX,0F07h { регистр ЦБ }
```

```
OUT DX,AX
```

```
MOV AX,0002h { регистр СЦ }
```

```
OUT DX,AX
```

```
MOV AL,CH
```

```
XOR AH,AH { значение функции возвращается }
```

```
{ в регистре AX }
```

```
end; { GetPixelAsm1 }
```

```
{*****}
```

Function GetPixelPas1(X,Y : Integer): word;

```
{ Режим чтения 1 }
```

```
{ используются регистры СЦ и ЦБ }
```

```
{ значение пикселя формируется по 1 биту; биты }
```

```
{ читаются из битовых плоскостей поочередно }
```

```
Var I : Integer;
```

```
R : registers;
```

```
ByteOfs : word;
```

```
begin
```

```
ByteOfs:=80 * Y + X shr 3;
```

```
{ байтовое смещение в ВП }
```



```

R.AH:= 1 shl ((X and 7) xor 7);
{ AH - битовая маска }

{ установка регистра P43 }
PortW[$3CE]:=$0805;
{ AH = режим записи 0 (биты 0,1) }
{ режим чтения 1 (бит 3) }
{ AL = индекс регистра }

{ установка регистра CЦ }
PortW[$3CE]:=$0F02;
{ AH = 1111B: значение регистра }
{ AL = индекс регистра }

R.CH:=0;
{ обнуляем накопитель значения пиксела }
R.BL:=7; { BL = индекс регистра ЦБ }
R.BH:=1;
{ BH = начальное значение регистра ЦБ }

{ (начинаем чтение с битовой плоскости 0) }
For I:=0 to 3 do
begin
  PortW[$3CE]:=R.BX;
  { установка регистра ЦБ }

```

```

{ Проверяем, установлен ли бит в пикселе }
{ для текущей битовой плоскости. Если }
{ установлен, то заносим бит в текущую }
{ позицию накопителя значения пиксела }
If Mem[$A000:ByteOfs] and R.AH <> 0 then
  R.CH:=R.CH or R.BH;

R.BH:=R.BH shl 1;
{ BH = номер следующей }
{ битовой плоскости }

end;
GetPixelPas1:=R.CH;
{ GetPixelPas1 = значение пиксела }

{ восстановление значений регистров по умолчанию }
PortW[$3CE]:=$0005;
{ регистр P43: режим записи 0, }
{ режим чтения 0 }
PortW[$3CE]:=$0F07; { регистр ЦБ }
PortW[$3CE]:=$0002; { регистр CЦ }

end; { GetPixelPas1 }

{*****}

```

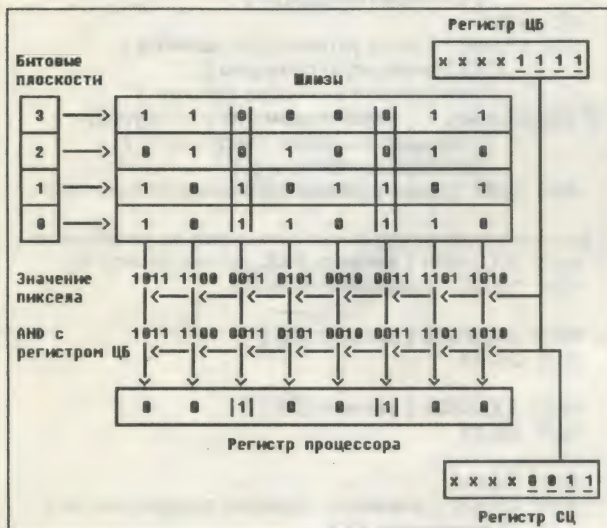


Рис. 4. Режим чтения 1 [2].

На результат сравнения влияет содержимое четырех младших битов (0—3) регистра ЦБ. Если какой-либо из битов установлен в 0, то соответствующий шлюз (0—3) исключается из сравнения. Например, значение 0011В (двоичное) указывает, что в сравнении принимают участие только два младших бита из значения каждого пиксела. А если в регистре записано 0000В, то значение в регистре CЦ не оказывает влияния на результат, и все биты в байте данных будут установлены в 1.

Режим записи 0 — самый сложный из всех. Он используется главным образом в тех случаях, когда необходимо изменить какой-либо один из восьми зашлюзованных пикселей. В формировании значения пиксела принимают участие регистры РУС, УС, БМ, ЦСД, а также байт данных процессора.

Из всего многообразия способов записи в режиме 0 можно выделить два основных, которым соответствуют крайние значения регистра РУС — 0 и 1111В. При первом способе (см. листинг 3) все четыре зашлюзованных байта перед записью в битовые плоскости комбинируются с байтом данных из регистра процессора (побайтовая комбинация, см. рис. 5а), при втором значение каждого зашлюзованного пиксела комбинируется с 4-битовым значением в регистре УС (попикселевая комбинация, см. рис. 5б). В обоих случаях на результат влияет содержимое регистров БМ и ЦСД. (Что касается промежуточных значений регистра РУС, то это весьма сложный вариант программирования, и используется он крайне редко).

Регистр БМ определяет судьбу каждого зашлюзованного пиксела. Если какой-либо бит регистра установлен в 0, то соответствующий пиксел без изменений пересылается из шлюзов в ВП. При установке бита в 1 значение пиксела перед пересылкой в ВП комбинируется либо с байтом данных процессора, либо со значением в регистре УС.

Способ же комбинирования задается регистром ЦСД, который выполняет две функции. В битах 3 и 4 этого регистра (младший бит имеет

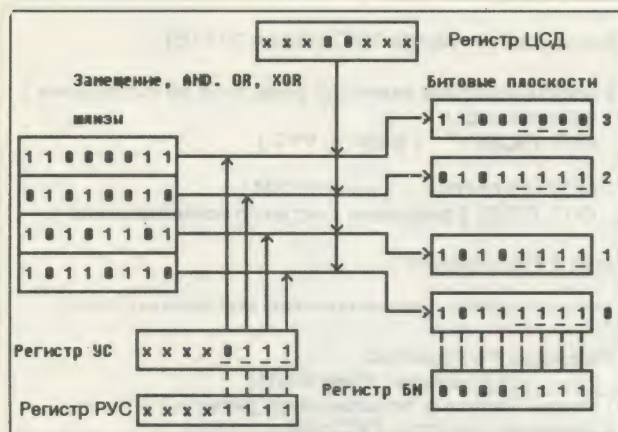


Рис. 5а. Режим записи 0. Значение в регистре БМ равно 00001111В, поэтому пиксели 0—3 копируются из регистра 4 С (т.к. значение в регистре РУС равно 1111В), а пиксели 4—7 копируются из шлюзов [2].

номер 0, старший — 7) указывается поразрядная логическая операция (AND, OR, XOR или просто замещение), выполняемая над байтом данных и зашлюзованными байтами (значение в регистре РУС равно 0, рис. 5б) или над значением в регистре УС и зашлюзованными пикселями (значение в регистре РУС равно 1111В, рис. 5а). Соответствие значений битов регистра ЦСД и выполняемых логических операций приведено в табл. 2.

Биты 0—2 регистра ЦСД определяют величину циклического сдвига вправо байта данных из регистра процессора перед комбинированием его с зашлюзованными байтами.

В режиме записи 0 при значении в регистре РУС, равном 1111В, в регистре УС фактически содержится четырехбитовое значение пикселя, которое пересылается в ВП; при этом содержимое регистра процессора не имеет значения. В том случае, когда необходимо изменить только один пиксел, в регистре БМ соответствующий бит устанавливается в 1 (остальные биты устанавливаются в 0), и выполняется операция записи в ВП. Если в регистре БМ все биты установить в 1, то за одну операцию записи в ВП будут изменены сразу все восемь пикселей, а это, как вы сами понимаете, уже горизонтальная линия.

Таблица 2. Соответствие значений битов 3 и 4 регистра ЦСД и поразрядных логических операций для режимов записи 0, 2 и 3 [1].

Бит 3	Бит 4	Функция
0	0	Замещение
1	0	AND
0	1	OR
1	1	XOR

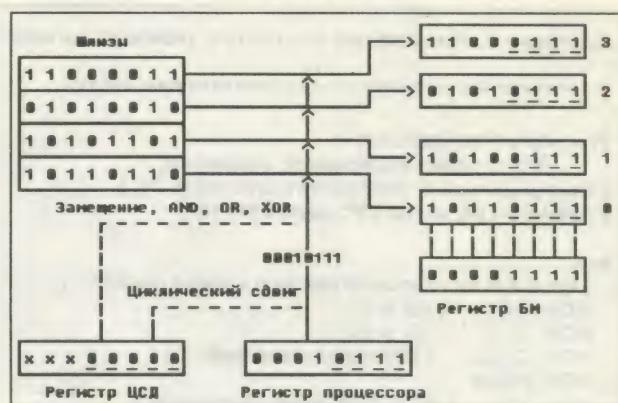


Рис. 5б. Режим записи 0. Значение в регистре БМ равно 00001111В, поэтому пиксели 0—3 копируются из регистра процессора (т.к. значение в регистре РУС равно 0000В), а пиксели 4—7 копируются из шлюзов [2].

Режим записи 1 представляет собой просто копирование содержимого шлюзов в битовые плоскости. На этот процесс не оказывает влияние ни один регистр. Этот режим используется, как правило, для пересылки байтов из одной области ВП в другую. Шлюзы загружаются по исходному адресу и копируются по адресу-приемнику.

Режим записи 2 (см. листинг 4) похож на режим записи 0 при значении регистра РУС 1111В, только здесь роль регистра УС выполняют четыре младших бита регистра процессора, а регистр РУС не влияет на результат (см. рис. 6). В остальном все происходит точно так же: битовые плоскости изменяются за счет комбинации значений пикселей в шлюзах со значением в регистре процессора (попикселевая комбинация) и выполняется логическая операция, установленная в регистре ЦСД, а регистр БМ определяет, какие пиксели изменяются, а какие нет.

Режим записи 3 поддерживается только на VGA и фактически представляет собой режим записи 0 при значении регистра РУС, равном 1111В, с одним лишь добавлением. Байт данных из регистра процессора циклически сдвигается вправо на число разрядов, заданных в регистре ЦСД, и комбинируется (операция AND) значением в регистре БМ. В результате получается битовая маска, которая играет такую же роль, как регистр БМ (см. рис. 7). (Пример для этого режима, к сожалению, отсутствует, поскольку у автора нет компьютера с адаптером VGA).

Все три режима записи могут к тому же быть модифицированы за счет использования регистра МП, о котором мы уже упоминали. Биты 0—3 этого регистра разрешают или запрещают

Листинг 3. Рисование пиксела в режиме записи 0 (значение регистра РУС равно 1111В).

```

{*****}
Procedure PutPixelAsm0
  (X,Y : Integer; Pixel: word); assembler;
{ Режим записи 0, используется регистр УС }
{ Значение регистра РУС равно 1111В }

asm
{ вычисление битовой маски и адреса пиксела }
MOV AX,Y { AX = Y }
MOV BX,X { BX = X }
MOV CH,BL { CH = младший байт X }
MOV DX,80
MUL DX { AX = Y*80 (байт на строку) }

MOV CL,3
SHR BX,CL { BX = X/8 }
ADD BX,AX { BX = байтовое смещение в ВП }

MOV AX,0A000h { AX = сегментный адрес ВП }
MOV ES,AX { ES:BX = байтовый адрес пиксела }

MOV CL,CH
AND CL,7
XOR CL,7 { битовая маска должна быть }
{ сдвинута на CL разрядов влево }
MOV AH,1 { AH = исходная битовая маска }

{ установка регистра БМ }
SHL AH,CL { AH = битовая маска в }
{ заданной позиции }
MOV DX,3CEH { DX = порт регистра ИР }
MOV AL,8 { AL = индекс регистра БМ }
OUT DX,AX

{ установка регистра УС }
MOV AH,byte(Pixel)
{ AH = значение в регистре УС }
MOV AL,0 { AL = индекс регистра }
OUT DX,AX

{ установка регистра РУС }
MOV AX,0F01h
{ AH = 0Fh: разрешены все битовые плоскости }
{ AL = 1: индекс регистра }
OUT DX,AX

{ установка значения пиксела }
OR ES:[BX],AL
{ загрузка шлюзов, их изменение и }
{ запись в ВП; значение регистра AL }
{ не влияет на результат }

{ восстановление значений регистров по умолчанию }
MOV AX,0001h
OUT DX,AX { регистр РУС }

MOV AX,0FF08h { регистр БМ }
OUT DX,AX { разрешен доступ ко всем пикселам }

end; { PutPixelAsm0 }
{*****}

Procedure PutPixelPas0
  (X,Y : Integer; Pixel: word);
{ Режим записи 0, используется регистр УС }
{ Значение регистра РУС равно 1111В }

Var R : registers;
    ByteOfs : word;

begin
  ByteOfs:=80 * Y + X shr 3;
    { байтовое смещение в ВП }

    { установка регистра БМ }
    R.AL:=8;
    R.AH:= 1 shl ((X and 7) xor 7);
    { AH = битовая маска }
    PortW[$3CE]:=R.AX;

    { установка регистра УС }
    R.AL:=0; { AL = индекс регистра }
    R.AH:=Pixel; { AH = значение пиксела }
    PortW[$3CE]:=R.AX;

    { установка регистра РУС }
    PortW[$3CE]:=$0F01;
    { AH = 0F: значение }
    { AL = 01: индекс регистра }

    { загрузка шлюзов, их изменение и запись в ВП; }
    { значение регистра AL не влияет на результат }
    Mem[$A000:ByteOfs]:=
      Mem[$A000:ByteOfs] or R.AL;

    { восстановление значений регистров }
    { по умолчанию }
    PortW[$3CE]:=$FF08; { регистр БМ }
    PortW[$3CE]:=$0001; { регистр УС }

end; { PutPixelPas0 }
{*****}

```

запись в битовые плоскости (точнее, в шлюзы). Если какой-либо бит установлен в 1, то соответствующего шлюза может быть изменено. Для того чтобы записать пиксел в ВП, необходимо выполнить следующие действия: загрузить чтением шлюзы из видеопамати, установить в позиции пиксела бит в регистре БМ (сформировать битовую маску), обнулить все биты в позиции пиксела в шлюзах, установить

биты в регистре МП (иными словами, поместить в него значение пиксела) и записать в шлюзы значение FFh.

В те шлюзы, которые разрешены для записи, будет записано значение 1 (а остальные мы уже обнулили, т.е. в шлюзах окажется значение пиксела). Затем содержимое шлюзов пересылается обратно в ВП (см. листинг 5).

Листинг 4. Рисование пиксела в режиме записи 2.

```

{*****}
Procedure PutPixelAsm2
  (X,Y : Integer; Pixel: word); assembler;
{ Режим записи 2 }

asm
{ вычисление битовой маски и адреса пиксела }
MOV AX,Y { AX = Y }
MOV BX,X { BX = X }
MOV CH,BL { CH = младший байт X }
MOV DX,80
MUL DX { AX = Y*80 (байт на строку) }

MOV CL,3
SHR BX,CL { BX = X/8 }
ADD BX,AX { BX = байтовое смещение в ВП }

MOV AX,0A000h { AX = сегментный адрес ВП }
MOV ES,AX { ES:BX = байтовый адрес пиксела }

MOV CL,CH
AND CL,7
XOR CL,7 { битовая маска должна быть }
{ сдвинута на CL разрядов влево }
MOV AH,1 { AH = исходная битовая маска }

{ установка регистра БМ }
SHL AH,CL { AH = битовая маска в }
{ заданной позиции }
MOV DX,3CEH { DX = порт регистра ИР }
MOV AL,8 { AL = номер регистра БМ }
OUT DX,AX

{ установка регистра P43 }
MOV AX,0205h
{ AH = режим записи 2 (биты 0,1) }
{ режим чтения 0 (бит 3) }
{ AL = индекс регистра }
OUT DX,AX

{ установка значения пиксела }
MOV AL,ES:[BX] { загрузка шлюзов из }
{ битовых плоскостей }
MOV AL,byte(Pixel)
{ значение пиксела пересылается }
{ в регистр процессора }
MOV ES:[BX],AL { запись в ВП }
{ восстановление значений регистров по умолчанию }

```

```

MOV AX,0005h { регистр P43: режим записи 0, }
OUT DX,AX { режим чтения 0 }

MOV AX,0FF08h { регистр БМ }
OUT DX,AX { разрешен доступ ко всем пикселаим }

end; { PutPixelAsm2 }

{*****}
Procedure PutPixelPas2(X,Y : Integer; Pixel: word);
{ Режим записи 2 }

Var R : registers;
    ByteOfs : word;

begin
  ByteOfs:=80 * Y + X shr 3;
    { байтовое смещение в ВП }

    { установка регистра БМ }
    R.AL:=8;
    R.AH:= 1 shl ((X and 7) xor 7);
    { AH = битовая маска }
    PortW[$3CE]:=R.AX;

    { установка регистра P43 }
    PortW[$3CE]:=$0205;
    { AH = режим записи 0 (бит 3) }
    { режим записи 2 (биты 0,1) }
    { AL = индекс регистра }

    { загрузка шлюзов байтами из битовых плоскостей }
    R.AL:=Mem[$A000:ByteOfs];

    { запись значения пиксела в ВП }
    Mem[$A000:ByteOfs]:=Pixel;

    { восстановление значений регистров по умолчанию }
    PortW[$3C5]:=$0005;
    { регистр P43 }
    { AH = режим чтения 0 (бит 3) }
    { режим записи 0 (биты 0,1) }
    PortW[$3CE]:=$FF08;
    { регистр БМ }
    { разрешен доступ ко всем пикселаим }

end; { PutPixelPas2 }

{*****}

```

СТАВИМ ТОЧКУ

Если вы сумели «продратся» через все эти биты, байты и регистры, то наверняка заметили, что нарисовать точку можно различными способами. Но в любом случае сохраняется общая логическая последовательность действий.

1. Вычисляется адрес ВП, где расположен нужный пиксел.

2. Устанавливаются все необходимые регистры.
3. Байты, находящиеся в битовых плоскостях по вычисленному адресу, загружаются в шлюзы.
4. В шлюзах устанавливается значение выбранного пиксела.
5. Байты из шлюзов пересылаются назад в битовые плоскости.

Фактически три последние операции могут быть выполнены одной командой процессора, что вы и видели в представленных процедурах.

В приведенных выше примерах регистр ЦСД остается установленным по умолчанию, хотя ло-

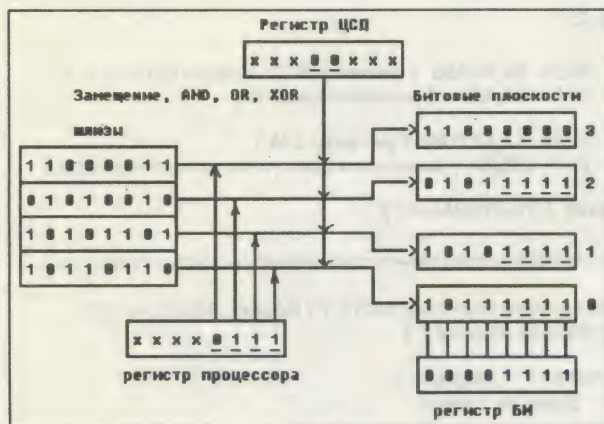


Рис. 6. Режим записи 2. Значение в регистре БМ равно 00001111В, поэтому пиксели 0—3 копируются из регистров процессора, а пиксели 4—7 копируются из шлюзов [2].

гическая операция XOR вызывает интересный эффект: при двух последовательных записях в ВП первоначальный цвет пиксела восстанавливается в исходный. Этот эффект можно использовать для рисования поверх изображения с последующим его восстановлением.

В тех случаях, когда требуемые значения регистров равны заданным по умолчанию, они, разумеется, не устанавливаются. Если же вы изменяете содержимое регистров, то не забудьте восстановить перед выходом из процедуры их исходные значения.

Для всех процедур были определены относительные скорости рисования точки, которые несколько различаются в зависимости от режима записи (естественно, процедуры на Паскале выполняются медленнее, чем на ассемблере). Если скорость выполнения стандартной процедуры

Листинг 5. Рисование пиксела в режиме записи 0 (значение регистра РУС равно 0000В).

```

{*****}

Procedure PutPixelAsm1
  (X,Y : Integer; Pixel: word); assembler;
{ Режим записи 0, используется регистр МП }
{ Значение регистра РУС равно 0000В }

asm
{ вычисление битовой маски и адреса пиксела }
MOV AX,Y { AX = Y }
MOV BX,X { BX = X }
MOV CH,BL { CH = младший байт X }
MOV DX,80
MUL DX { AX = Y*80 (байт на строку) }

MOV CL,3
SHR BX,CL { BX = X/8 }
ADD BX,AX { BX = байтовое смещение в ВП }

MOV AX,0A000h { AX = сегментный адрес ВП }
MOV ES,AX { ES:BX = байтовый адрес пиксела }

MOV CL,CH
AND CL,7
XOR CL,7 { битовая маска должна быть }
{ сдвинута на CL разрядов влево }
MOV AH,1 { AH - исходная битовая маска }

{ установка регистра БМ }
SHL AH,CL { AH = битовая маска }
{ в заданной позиции }
MOV DX,3CEH { DX = порт регистра ИР }
MOV AL,8 { AL = номер регистра БМ }
OUT DX,AX

MOV AL,ES:[BX] { загрузка шлюзов байтами }
{ из битовых плоскостей }
MOV byte ptr ES:[BX],0
{ установили в 0 все биты пиксела }

{ установка регистра МП }
MOV DL,0C4H { DX = порт адресного регистра }

{ блока синхронизации }
MOV AH,byte(Pixel)
{ AH = значение в регистре; }
{ биты, установленные в 1, разрешают }
{ запись в битовые плоскости }
MOV AL,2 { AL = индекс регистра МП }
OUT DX,AX

MOV byte ptr ES:[BX],0FFH
{ установка битов в разрешенные }
{ битовые плоскости (т.е. в ВП) }
{ записывается значение пиксела }

{ восстановление значения регистра МП }
{ по умолчанию }
MOV AH,0FH { разрешен доступ ко всем битовым }
OUT DX,AX { плоскостям }

{ восстановление значения регистра БМ }
{ по умолчанию }
MOV DL,0CEH { DX = 3CEH }
MOV AX,0FF08H { разрешен доступ }
OUT DX,AX { ко всем пикселам }
end; { PutPixelAsm1 }

{*****}

Procedure PutPixelPas1
  (X,Y : Integer; Pixel: word);
{ Режим записи 0, используется регистр МП }
{ Значение регистра РУС равно 0000В }

Var R : registers;
    ByteOfs : word;

begin
  ByteOfs:=80 * Y + X shr 3;
  { байтовое смещение в ВП }

  { установка регистра БМ }
  R.AL:=8;

```



```

R.AH:= 1 shl ((X and 7) xor 7);
{ AH = битовая маска }
PortW[$3CE]:=R.AX;
{ загрузка шлюзов байтами из битовых }
{ плоскостей и обнуление всех битов пиксела }
Mem[$A000:ByteOfs]:=Mem[$A000:ByteOfs] and 0;

{ установка регистра МП }
R.AH:=byte(Pixel);
{ AH = значение в регистре; }
{ биты, установленные в 1, разрешают }
{ запись в битовые плоскости }
R.AL:=2; { AL = индекс регистра }
PortW[$3C4]:=R.AX;

{ установка битов в разрешенные битовые }
{ плоскости, т.е. в ВП записывается }

```

```

{ значение пиксела }
Mem[$A000:ByteOfs]:=$FF;

{ восстановление значений регистров по умолчанию }
PortW[$3C4]:=$0F02;
{ регистр МП: }
{ разрешен доступ }
{ ко всем битовым плоскостям }
PortW[$3CE]:=$FF08;
{ регистр БМ: }
{ разрешен доступ }
{ ко всем пикселям }

end; { PutPixelPas1 }

{*****}

```

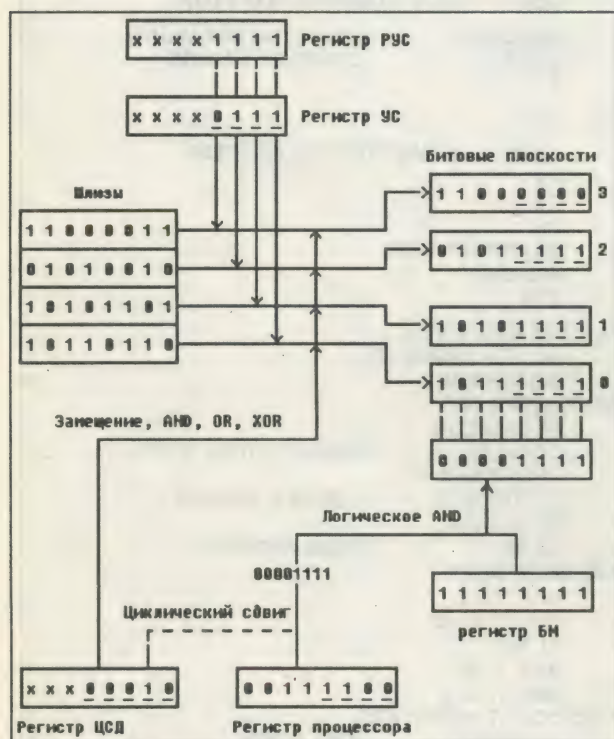


Рис. 7. Режим записи 3 — только для адаптера VGA [2].

PutPixel из библиотеки GRAPH.TPU принять за 1, то остальные процедуры выстроятся в следующий ряд по возрастанию:

```

PutPixel : PutPixelPas1 : PutPixelPas0 : PutPixelPas2 :
PutPixelAsm1 : PutPixelAsm0 : PutPixelAsm2 =
1 : 1.42 : 1.46 : 1.55 : 2.32 : 2.32 : 2.55

```

Как видим, даже процедуры, написанные на Паскале, но использующие «прямой» доступ к видеопамяти, выполняются быстрее, чем стандартная PutPixel. А наивысшая скорость рисования принадлежит процедуре PutPixelAsm2, использующей режим записи 2.

Такой же ряд можно привести и для функций определения значения пиксела:

```

GetPixelPas1 : GetPixelPas0 : GetPixel : GetPixelAsm1 :
GetPixelAsm0 = 0.87 : 0.93 : 1 : 1.77 : 2.12

```

Здесь паскалевские процедуры выполняются медленнее стандартной GetPixel, но с ассемблерными все обстоит благополучно.

Сходная картина наблюдается и при рисовании других графических примитивов, так что время, потраченное вами на изучение принципов программирования видеоадаптера не пропадет зря (устройство видеоадаптеров, алгоритмы рисования графических примитивов и многое другое вы можете найти в [2] и [3]).

В заключение хотелось бы еще раз напомнить, что все процедуры и функции работают только в основных графических режимах видеоадаптеров EGA и VGA.

Литература

1. IBM Enhanced Graphics Adapter. Technical Reference.
2. Wilton, R. Programmer's Guide to PC & PS/2 Video Systems. — NY: Microsoft Press, 1987. — 533 p.
3. Kliewer, B.D. EGA/VGA Programmer's Reference Guide. — NY: Intertext Publications, 1988. — 269 p.

Здравствуйте!

Мне, да и не только мне, хотелось бы узнать, существует ли в каком-либо языке программирования возможность включения режима Turbo или просто изменения тактовой частоты работы компьютера?

Если есть такая возможность, то где, в каких языках и как этого можно добиться? Если же такой возможности нет, я думаю, что будущим разработчикам языков или их новых версий следует задуматься об этом...

Сенных Александр, г. Екатеринбург



Вопрос управления тактовой частотой процессора связан не только с программированием, но и с аппаратным обеспечением. В компьютерах PC XT и PC AT корпорации IBM такой возможности предусмотрено не было, поэтому долгое время промышленного стандарта на аппаратное устройство системы управления тактовой частотой процессора не существовало. Некоторые фирмы, конечно, встраивали подобные системы в свои компьютеры, однако широкого распространения (как и широкой программной поддержки) они не получили. Положение изменилось лишь недавно — был разработан стандарт Advanced Power Management (APM), предусматривающий управление тактовой частотой процессора в целях снижения потребляемой компьютером мощности. Прочтите о нем вы можете в статье «Портативные компьютеры приближают будущее» в «Мир ПК», № 10/92. Речь там идет о новейших моделях портативных компьютеров.

Однако существуют и чисто программные средства управления скоростью работы компьютера, реализованные в множестве программ, например в утилите GOSLOW фирмы Strange Software, Ltd. Подобные программы перехватывают прерывания системного таймера компьютера и часть времени занимают процессор бесполезной работой. Однако работа компьютера замедляется только в те моменты, когда разрешены прерывания, т. е. не всегда.

Листинг 1.

```

DMA1PG EQU      83H
BAR_T EQU       2
WCR_T EQU       3
DCON EQU        8
DBITS EQU       0AH ;SINGLE MASK BIT REGISTER
DMODE EQU       0BH ;DMA MODE REG
CFLOP EQU       0CH ;BYTE COUNTER FLIP-FLOP

DMACONEQU       0000000B ;DMA CONTROL WORD
TMODE EQU       01011001B ;CHANNEL 1 MODE
EN1 EQU         1

                ORG      100H

BEG: SUB        AX,AX ;STORE RETURN ADDRESS
      PUSH      AX
      PUSH      CS
      POP       DS
      MOV       DX,OFFSET MSG
      MOV       AH,09H
      INT       21H

;PROGRAMMING DMA PAGE REGISTERS
      MOV       AL,0FH ;PAGE=F
      OUT       DMA1PG,AL

;PROGRAMMING DMA CONTROLLER
      MOV       AL,DMACON
      OUT       DCON,AL ;DMA CONTROL WORD
      JMP       SHORT $+2
      OUT       CFLOP,AL ;CLEAR FLIP-FLOP
      JMP       SHORT $+2
      MOV       AX,00 ;BASE ADDRESS

;OUTPUT TO DMA BASE ADDR REGS
      OUT       BAR_T,AL
      JMP       SHORT $+2
      MOV       AL,AH
      OUT       BAR_T,AL
      JMP       SHORT $+2

;OUTPUT TO DMA WORD COUNT REGISTERS
      MOV       AX,4000H ;# OF BYTES
      OUT       WCR_T,AL
      JMP       SHORT $+2
      MOV       AL,AH
      OUT       WCR_T,AL
      JMP       SHORT $+2

;output to DMA mode register
      MOV       AL,TMODE
      OUT       DMODE,AL
      JMP       SHORT $+2

;OPEN DMA CHANNEL
      MOV       AL,EN1
      OUT       DBITS,AL
      RET

;THAT'S ALL
MSG: DB          0AH,0DH," Now your computer!"
      DB          " has XT speed",0AH,0DH,"$"
      END

```


Листинг 2.

```

DBITS EQU 0AH ;MASK BIT REGISTER
DE1 EQU 5
ORG 100H
;PROGRAM BEGINNING
BEG: SUB AX,AX ;STORE RETURN ADDRESS
      PUSH AX
      PUSH CS
      POP DS
      MOV DX,OFFSET MSG
      MOV AH,09H
      INT 21H
;CLOSE DMA CHANNEL
      MOV AL,DE1
      OUT DBITS,AL
      RET
MSG: DB 0AH,0DH," Now your computer"
      DB " has AT speed",0AH,0DH,"$"
      END

```

Однажды я занимался отладкой контроллера для PC AT, в котором для связи с внешними устройствами используется прямой доступ к памяти (ПДП). Забыв вставить контроллер в разъем расширения компьютера, я запустил драйвер контроллера. К моему удивлению, компьютер не «завис», а стал работать медленнее (чуть быстрее PC XT). Играть в таком режиме в DIGGER — одно удовольствие.

По «мотивам» драйвера контроллера я впоследствии написал программу LIKE-XT.COM, исходный текст которой приведен в Листинге 1. Програм-

ма LIKE-XT.COM работает не со всеми ПК. Это зависит от уровня сигнала на входе DRQ1 (вход запроса прямого доступа к каналу 1 контроллера ПДП) на разъеме системной платы. Программа LIKE-XT.COM настраивает контроллер ПДП компьютера на передачу данных из памяти во внешнее устройство. Если на выводе DRQ1 присутствует логическая единица (когда на системной плате есть резистор между входом DRQ1 и питанием), программа замедляет компьютер, если логический ноль — ничего не происходит. Замедление работы компьютера происходит за счет

того, что после каждого цикла обращения процессора к памяти происходит цикл прямого доступа. Такое замедление является аппаратным и не зависит от состояния флажка разрешения прерываний. Из четырех имеющихся в PC AT восьмибитовых каналов прямого доступа я задействовал канал 1. Этот канал обычно свободен, а использоваться он может сканером или сетевым контроллером.

После запуска программы LIKE-XT.COM компьютеры иногда перестают работать с НГМД. Это связано с тем, что для обмена с гибким диском используется канал 2 ПДП, а он имеет более низкий приоритет, чем канал 1. Чтобы исправить это, можно перехватить прерывание и на моменты обращения к НГМД запрещать прямой доступ к памяти через канал 1. Для тех случаев, когда нужно без перезагрузки восстановить нормальную работу компьютера, я написал программу LIKE-AT.COM. Ее исходный текст приведен в Листинге 2. Если к сигналу DRQ1 (вывод B18 разъема ISA) подключить тумблер, можно будет управлять режимом Turbo не только программно, но и вручную.

И.Б. Рожожкин

Читатель советует

Некоторые из поступающих в редакцию писем читателей содержат не вопросы, а замечания и ответы на ранее опубликованные в журнале вопросы. Это письмо является откликом на опубликованные в «Мире ПК», № 4/1992 на стр. 128-129 вопрос и ответ, касающиеся несовместимости дискет емкостью 360 Кбайт, размеченных на дисководах 360 Кбайт и 1,2 Мбайт. Мне также приходилось сталкиваться с этой проблемой, и я, по-видимому, смогу дать совет, как поступить, если нет возможности ни

купить другой дисковод, ни, тем более, отказаться от использования старого компьютера XT.

В ответе неточно указана причина несовместимости: в действительности дело здесь не в различной силе магнитного сигнала, наводимого в головках чтения/записи дисковода, а в ширине головки, которая у накопителя на 360 Кбайт вдвое больше, чем у накопителя 1,2 Мбайт; из-за этого компьютер XT, на котором установлен накопитель на 360 Кбайт, пытается считывать не только саму дорожку, но и широкую полосу

вокруг нее, что и приводит к ошибкам. Чтобы добиться устойчивого чтения на накопителе 360 Кбайт дискет, записанных на дисководе 1,2 Мбайт, мы с коллегами применяем следующий способ.

Дискета размагничивается с помощью либо постоянного магнита (необходимо несколько раз провести по ней магнитом — естественно, через конверт), либо (это несколько лучше) соленоида и форматруется на дисководе 1,2 Мбайт. После этого информация, записанная на компьютере AT (с дисково-

дом 1,2 Мбайт), как показывает опыт, читается на компьютере XT (с дисководом 360 Кбайт) вполне устойчиво. Естественно, допустимо только чтение — при попытке записать или удалить на дисководе 360 Кбайт какой-либо файл дорожки немедленно сделаются широкими и дискета станет непригодной для переноса информации с AT на XT. Поэтому для обратного переноса следует применять другую дискету.

В.А. Горпенко

Contents

IN FOCUS	<i>You, the Future, and Mobile Computing</i>	6		<i>Logical Games Life Story</i>	60
	KATHLEEN WIEGNER			IGOR GRIGORYEV	
HARDWARE	<i>Nomads of the Nineties</i>	12	COMPANY	<i>Intel in Moscow</i>	64
	JEFFREY ROTHFEDER		PROFILE	IVAN B. ROGOJKINE	
	<i>Good Things Come in Small Packages</i>	17	WINDOWS	<i>The Mysterious PIF Editor</i>	70
	ANITA AMIRREZVANI			BRIAN LIVINGSTON	
	<i>Mobile Computing at Five Miles per Second</i>	19	FORUM	<i>«KEYMACRO.LEX-92» Contest Total</i>	79
	PHILIP CHIEN			GRIGORY SHMERLING	
	<i>Pioneers of the Information Age</i>	20		<i>Dark News of Dark Avenger</i>	82
	BRYAN HASTINGS			BRYAN BRANDENBURG, EVGENY V. KASPERSKY	
	<i>Holiday Buyers' Guide</i>	22		<i>Amateur and Professional</i>	87
	ANITA AMIRREZVANI			IVAN B. ROGOJKINE	
	<i>A Visual Glossary of Mobile Computing</i>	28	YOUR CHOICE	<i>Time Line 5.0: Now with Graphics</i>	89
	STEVE BASS			SERGE GLADKOFF	
COMPUTER GAMES	<i>Show It to be Seen</i>	31	SOFTWARE	<i>C and Pascal Compatible Library</i>	97
	MICHAEL DONSKOY			VADIM A. NECHAYEV, ANDREY K. MALYARSKY	
	<i>Animation in Computer Games</i>	33		<i>Borland Pascal 7.0: New Features</i>	105
	ALEKSEY ORLOV			VALERY V. FARONOV	
	<i>My Own Game</i>	42	PRACTICAL WORK	<i>To Put a Pixel</i>	115
	ANDREY RODIONOV			VYACHESLAV G. CHERTKOV	
	<i>Do Games Bring Success? Yes, if You are Lucky Georgy Pachikov's Interview</i>	56		<i>Letters</i>	126, 127
				<i>News</i>	59, 114



«Мир ПК» — издание International Data Group, крупнейшей в мире информационной корпорации по вычислительной технике. International Data Group выпускает более чем в 50 странах мира 150 изданий по компьютерной тематике. 30 миллионов человек ежемесячно читают журналы и другие публикации International Data Group.

Представляем издания International Data Group: Австралия — Computerworld Australia, Australian PC World, Australian Macworld, Profit, Information Decisions, Reseller; Австрия — Computerwelt Oesterreich; Азия — Computerworld Hong Kong, Computerworld Southeast Asia, Computerworld Malaysia; Аргентина — Computerworld Argentina, InfoWorld Argentina; Болгария — Computerworld Bulgaria; Бразилия — DataNews, PC Mundo, Mundo IBM, Mundo Unix, Automacao & Industria, Publish; Великобритания — Lotus, MacWorld; Венгрия — Computerworld SZT, Microvilag Magazin, PC Vilag; Венесуэла — Computerworld Venezuela, MicroComputerworld Venezuela; Германия — Computerwoche, Computerwoche Focus, Computerwoche Extra, Computerwoche Karriere, Information Management, MacWelt, NetzWelt, OS/2 Welt, PC Woche, PC Welt, Unix Welt, Unit, Lotus Welt; Греция — PC World; Дания — CAD/CAM WORLD, Computerworld Danmark, PC World Danmark, MacWorld Danmark, Computerworld Focus, Lotus World, Macintosh Produktkatalog, Unix World, PC/LAN World; Египет — PC World Middle East; Израиль — People & Computers; Индия — Computers & Communications; Испания — CIMWorld, Comunicaciones World, Computerworld Espana, PC World Espana, MacWorld, PC World Autoedicion, Amiga World, Publish; Италия — Computerworld Italia, PC World Italia, MacWorld Italia, Network World Italia; Канада — ComputerData, Direct Access, Graduate Computerworld; КНР — China Computerworld, PC World China; Ко-

лумбия — Computerworld Columbia; Мексика — Computerworld Mexico, PC Journal; Нигерия — PC World Africa; Нидерланды — Computerworld Netherlands, LAN Magazine, Mac Magazine, Computer! Totaal; Новая Зеландия — Computerworld, PC World; Норвегия — Computerworld Norge, PC World, PC World Norge, PC World Express, IDG Direct Response, Multimedia and Desktop, Lotus World, PC World's Product Guide, Student's DP-Guide, Publish! World, Macworld Norge; Перу — PC World; Польша — Computerworld Poland; Россия — Мир ПК, КомпьютерУолд, Сети, Мир ПК-Диск; Румыния — Infoclub; США — Amiga World, CIO, Computerworld, Computer Buyers World, Digital News, Electronic News, Federal Computer Week, GamePro, inCider/A+, IDG Books, InfoWorld, Lotus, MacWorld, MPC World, NeXTWORLD, Network World, PC Games, PC World, Portable Office, PC Letter, Publish!, Run, SunWorld; Таиланд — Thai Computerworld; Тайвань — Computerworld Taiwan, PC World Taiwan; Турция — Computerworld, PC World; Филиппины — Computerworld, PC World; Финляндия — Mikro PC, Tietoviikko, Tietotekniikka, Tietoverikko; Франция — Le Monde Informatique, Distributique, Compu Search, Golden, Computer Direct, InfoPC, Telecoms International, Le Guide du Monde Informatique; Чехо-Словакия — Computerworld Czechoslovakia, PC World Czechoslovakia, Network World, Nural; Чили — Informatica; Швейцария — Computerworld Schweiz, MacWorld Schweiz; Швеция — ComputerSweden, Microdatom, MacWorld, CAD/CAM World, Lotus, Windows, Svenska PC World, Lokala Natverk/LAN, Affarsekonomi Management, Attack, CAP, Datangenjoren, Data & Telekommunikation, Maxi Data, Digital/Varlden, Unix; Югославия — Moj Mikro; Южная Корея — Computerworld Korea, PC World Korea; Япония — Computerworld Japan, MacWorld Japan; IDG HIGH Tech — Newproductworld.

ComputerLand

MBL International East Inc. d/b/a Computerland -
официальный дилер корпорации Intel -
представляет средства
расширения возможностей
персональных компьютеров.

intel

Ваши компьютеры

Intel 14.4kx Modem.
The fastest way
to save money.

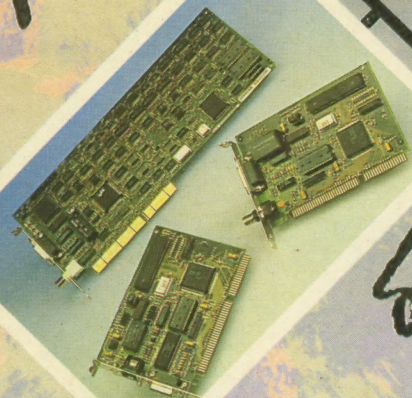


The Intel
SatisFAXtion
Board
is the easiest
way to fax.
Period.

Intel SatisFAXtion Modem/400
The best modem. The best fax. Communicate with everyone.

Intel SatisFAXtion Modem/200
Great looking faxes. A cinch to use.

Intel SatisFAXtion
Modem/100
Dialing Fax. Great software.



*Будут
работать
эффективнее!*

- Математические и инженерные сопроцессоры
- Сетевое оборудование
- Процессоры OverDrive™
- Модемы
- Факс-модемы

121248, Москва, Кутузовский пр-т, 8
Тел. (095) 243-35-53, факс (095) 243-78-48 (Магазин-салон)
125027, Санкт-Петербург, Свердловская наб., 54
Тел. (812) 224-09-32, факс (812) 224-29-28 (Магазин-салон)
820075, Екатеринбург, ул. Луначарского, 81
Тел. (343) 244-19-58, факс (343) 255-73-46

252021, Киев, ул. Грушевского, 30/1
Тел. (044) 293-10-03, факс (044) 293-10-03
232698, Вильнюс, ул. Вильнюс, 22 (Магазин-салон)
Тел. (012) 261-83-55/261-83-64, факс (012) 222-39-32
404111, Волжский, Волгоградской обл., ул. Сталинградская, 8
Тел. (844) 597-33-47/597-30-12, факс (844) 597-32-83

614600, Пермь, ул. Крылова, 36
Тел. (34-22) 25-83-47, факс (34-22) 25-89-88
183000, Мурманск, ул. Книповича, 23а
Тел. (815-00) 5-65-88, факс (815-00) 5-65-88
375001, Ереван, ул. Саят-Нова, 19
Тел. (885) 256-26-86

ComputerLand®

Business to business. Person to person.

The logo for ZEOS International, Ltd. is presented on a red rectangular background that is tilted slightly to the right. The word "ZEOS" is written in a large, white, stylized serif font, with a registered trademark symbol (®) positioned at the top right of the letter 'S'. Below "ZEOS", the words "INTERNATIONAL, LTD." are written in a smaller, white, sans-serif font, following the same tilt as the background.

ZEOS[®]

INTERNATIONAL, LTD.

НАИЛУЧШИЕ КОМПЬЮТЕРЫ АМЕРИКИ.

**ТОРГОВЫЕ ПРЕДСТАВИТЕЛЬСТВА И
ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ**

Международный Компьютерный Клуб (МКК)

РОССИЯ, Москва 101813
Центр, проезд Серова д. 4,
"Знание"
Тел: Москва 7-095-921-09-02
Телекс: 411630
Факс: Москва 7-095-921-09-02

В США
ZEOS International, Ltd.
530 5th Avenue N.W.
St. Paul, MN 55112
U.S.A.
Tel: 7-095-612-633-6131 Fax: 7-095-612-633-6131
7-812-612-633-1325 7-812-612-633-1325